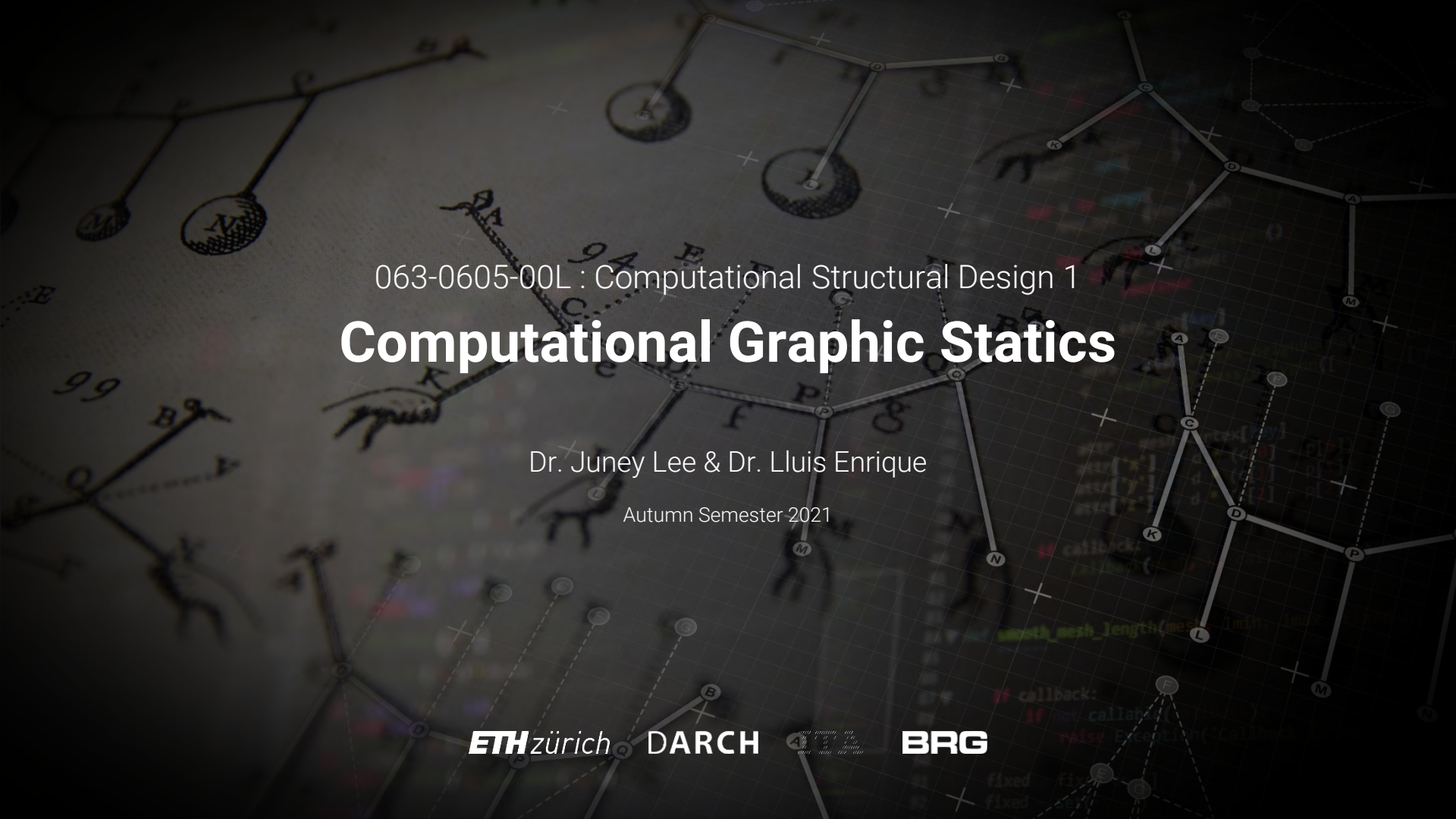


063-0605-00L : Computational Structural Design 1

Computational Graphic Statics

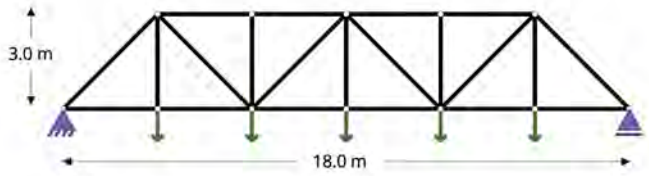
Dr. Juney Lee & Dr. Lluis Enrique

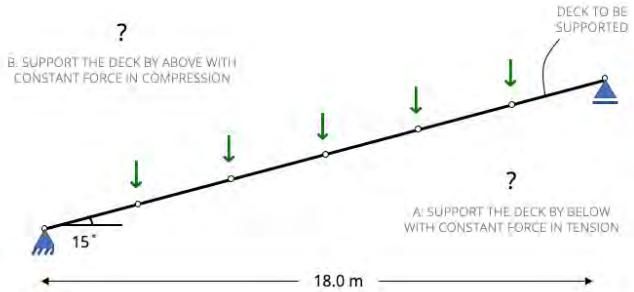
Autumn Semester 2021

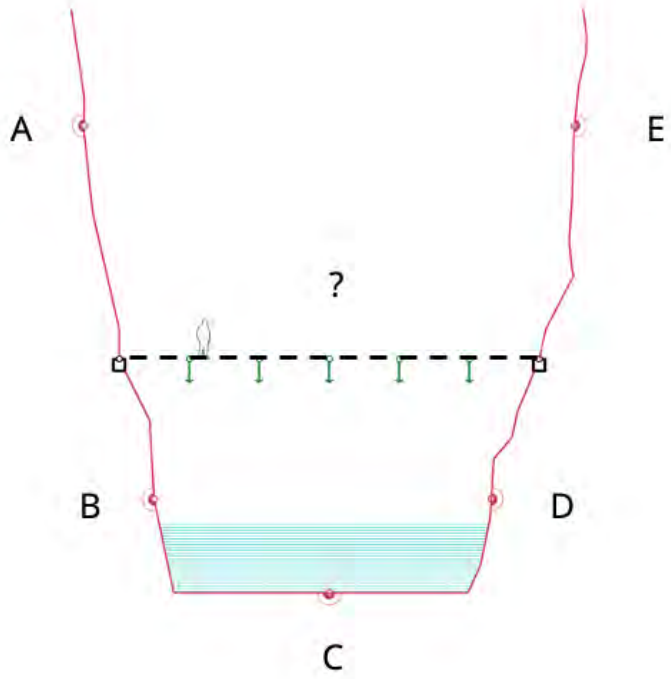
ETH zürich DARCH BRG

Week 9 Friday, November 19th

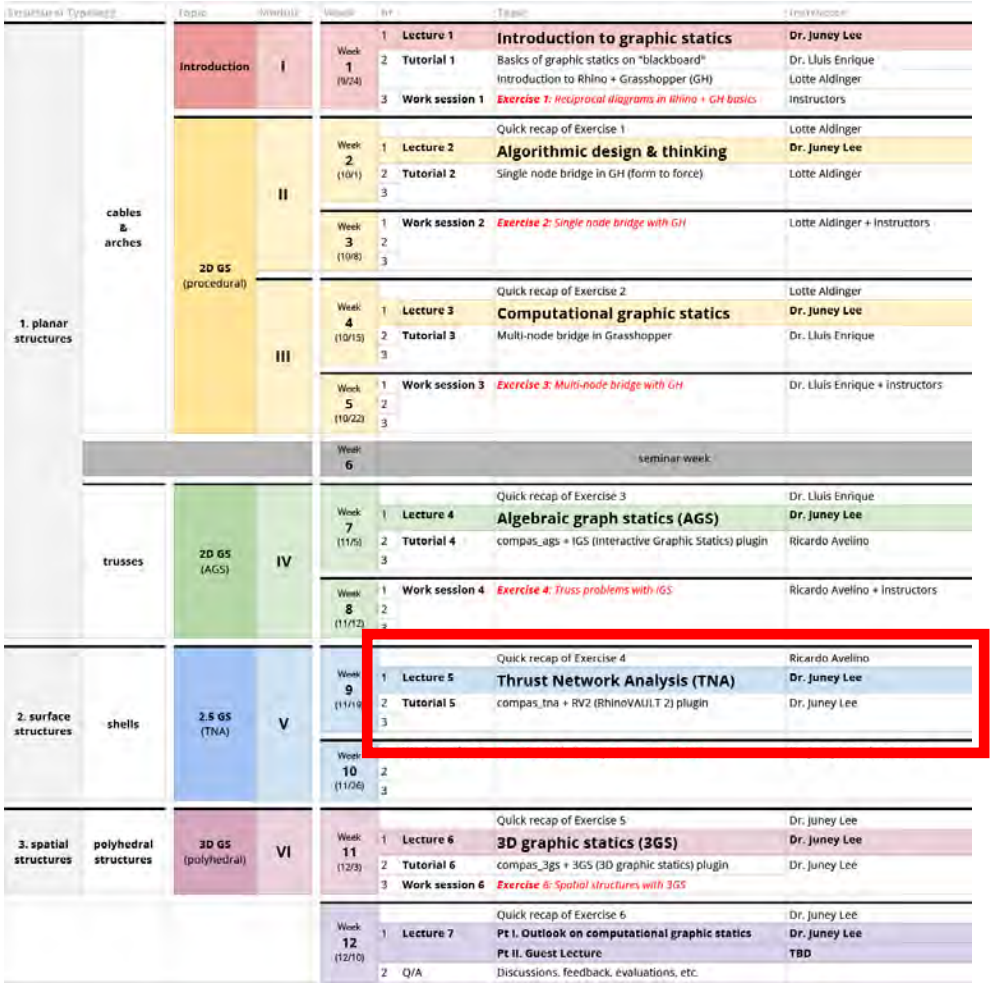
Hour 1	**09:45 – 09:50**	· Recap of Exercise 4
	09:50 – 10:30	· **Lecture 4 : Thrust Network Analysis**
	10:30 – 10:45	· Break
Hour 2	**10:45 – 11:35**	· **Tutorial 4 : RhinoVAULT 2 (RV2) – Part 1**
	11:30 – 11:45	· Break
Hour 3	**11:45 – 12:30**	· **Tutorial 4 : RhinoVAULT 2 (RV2) – Part 2**


3.0 m
18.0 m


?
B: SUPPORT THE DECK BY ABOVE WITH CONSTANT FORCE IN COMPRESSION
DECK TO BE SUPPORTED
?
A: SUPPORT THE DECK BY BELOW WITH CONSTANT FORCE IN TENSION
15°
18.0 m

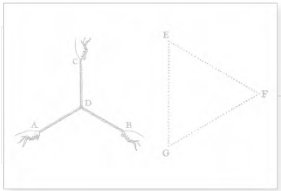

A
E
?
B
D
C

Course schedule



Structural Typology	Topic	Workflow	Week	Nr	Topic	Instructor
1. planar structures	cables & arches	Introduction I	Week 1 (9/24)	1	**Lecture 1**	**Introduction to graphic statics**	**Dr. Juney Lee**
				2	**Tutorial 1**	Basics of graphic statics on "blackboard"	Dr. Lluis Enrique
						Introduction to Rhino + Grasshopper (GH)	Lotte Aldinger
				3	**Work session 1**	***Exercise 1**: Reciprocal diagrams in Rhino + GH basics*	Instructors
		2D GS (procedural) II	Week 2 (10/1)			Quick recap of Exercise 1	Lotte Aldinger
				1	**Lecture 2**	**Algorithmic design & thinking**	**Dr. Juney Lee**
				2	**Tutorial 2**	Single node bridge in GH (form to force)	Lotte Aldinger
				3			
			Week 3 (10/8)	1	**Work session 2**	***Exercise 2**: Single node bridge with GH*	Lotte Aldinger + instructors
				2			
				3			
		III				Quick recap of Exercise 2	Lotte Aldinger
			Week 4 (10/15)	1	**Lecture 3**	**Computational graphic statics**	**Dr. Juney Lee**
				2	**Tutorial 3**	Multi-node bridge in Grasshopper	Dr. Lluis Enrique
				3			
			Week 5 (10/22)	1	**Work session 3**	***Exercise 3**: Multi-node bridge with GH*	Dr. Lluis Enrique + instructors
				2			
				3			
			Week 6			seminar week	
	trusses	2D GS (AGS) IV	Week 7 (11/5)			Quick recap of Exercise 3	Dr. Lluis Enrique
				1	**Lecture 4**	**Algebraic graph statics (AGS)**	**Dr. Juney Lee**
				2	**Tutorial 4**	compas_ags + IGS (Interactive Graphic Statics) plugin	Ricardo Avelino
				3			
			Week 8 (11/12)	1	**Work session 4**	***Exercise 4**: Truss problems with IGS*	Ricardo Avelino + instructors
				2			
				3			
2. surface structures	shells	2.5 GS (TNA) V	Week 9 (11/19)			Quick recap of Exercise 4	Ricardo Avelino
				1	**Lecture 5**	**Thrust Network Analysis (TNA)**	**Dr. Juney Lee**
				2	**Tutorial 5**	compas_tna + RV2 (RhinoVAULT 2) plugin	Dr. Juney Lee
				3			
			Week 10 (11/26)	2			
				3			
3. spatial structures	polyhedral structures	3D GS (polyhedral) VI	Week 11 (12/3)			Quick recap of Exercise 5	Dr. Juney Lee
				1	**Lecture 6**	**3D graphic statics (3GS)**	**Dr. Juney Lee**
				2	**Tutorial 6**	compas_3gs + 3GS (3D graphic statics) plugin	Dr. Juney Lee
				3	**Work session 6**	***Exercise 6**: Spatial structures with 3GS*	
			Week 12 (12/10)			Quick recap of Exercise 6	Dr. Juney Lee
				1	**Lecture 7**	**Pt I. Outlook on computational graphic statics**	**Dr. Juney Lee**
						Pt II. Guest Lecture	**TBD**
				2	Q/A	Discussions, feedback, evaluations, etc.	

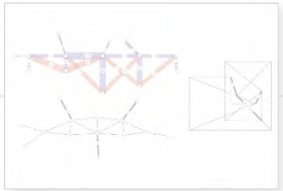
2D



Module I
Basics of Graphic Statics

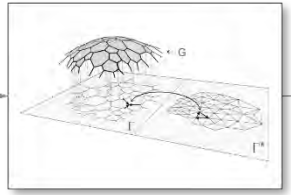
Module II
Interactive Graphic Statics I (single node)

Module III
Interactive Graphic Statics 2 (multi node)



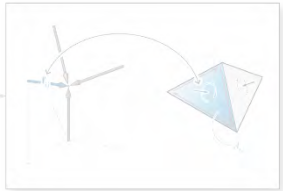
Module IV
Algebraic Graph Statics

2.5D



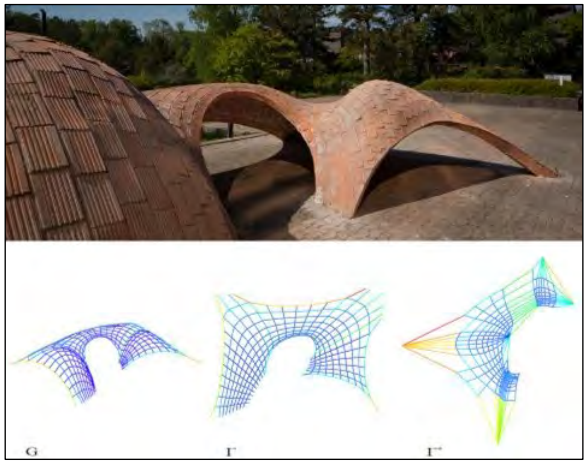
Module V
Thrust Network Analysis

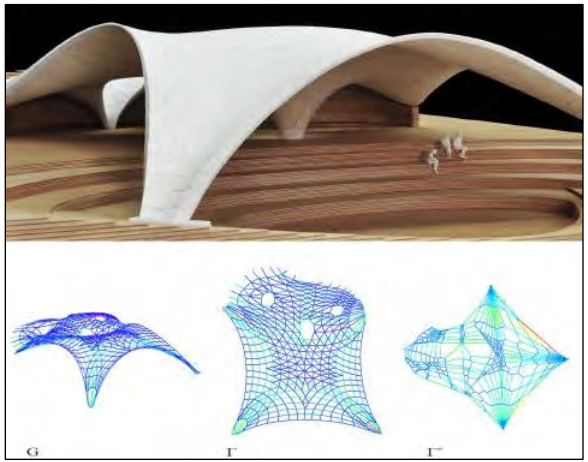
3D

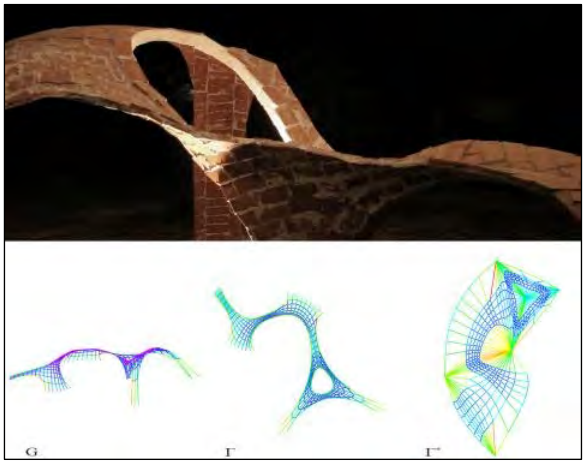


Module VI
3D Graphic Statics

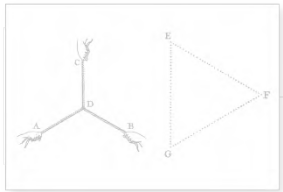
Module V · Thrust Network Analysis & RhinoVAULT







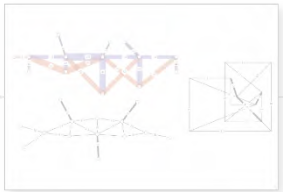
2D | 2.5D | 3D



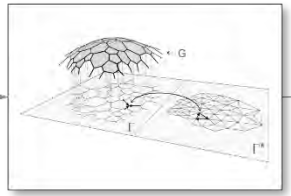
Module I
Basics of Graphic Statics

Module II
Interactive Graphic Statics I (single node)

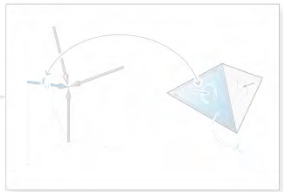
Module III
Interactive Graphic Statics 2 (multi node)



Module IV
Algebraic Graph Statics

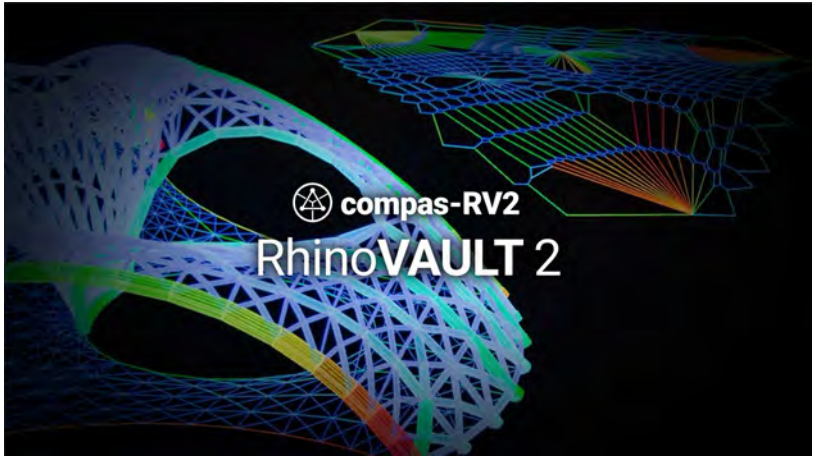


Module V
Thrust Network Analysis



Module VI
3D Graphic Statics

Module V · Thrust Network Analysis & RhinoVAULT



063-0605-00L : Computational Structural Design 1

Computational Graphic Statics

Lecture 5
Thrust Network Analysis

Friday, November 19th, 2021

Dr. Juney Lee

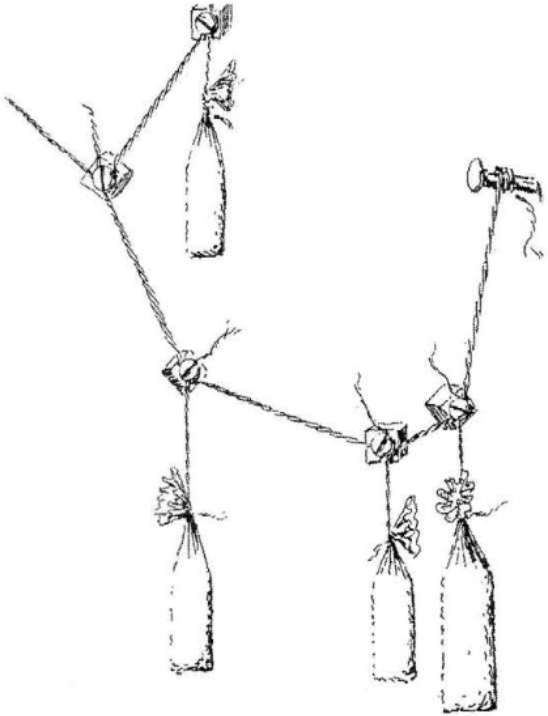
ETH zürich DARCH BRG

Form finding

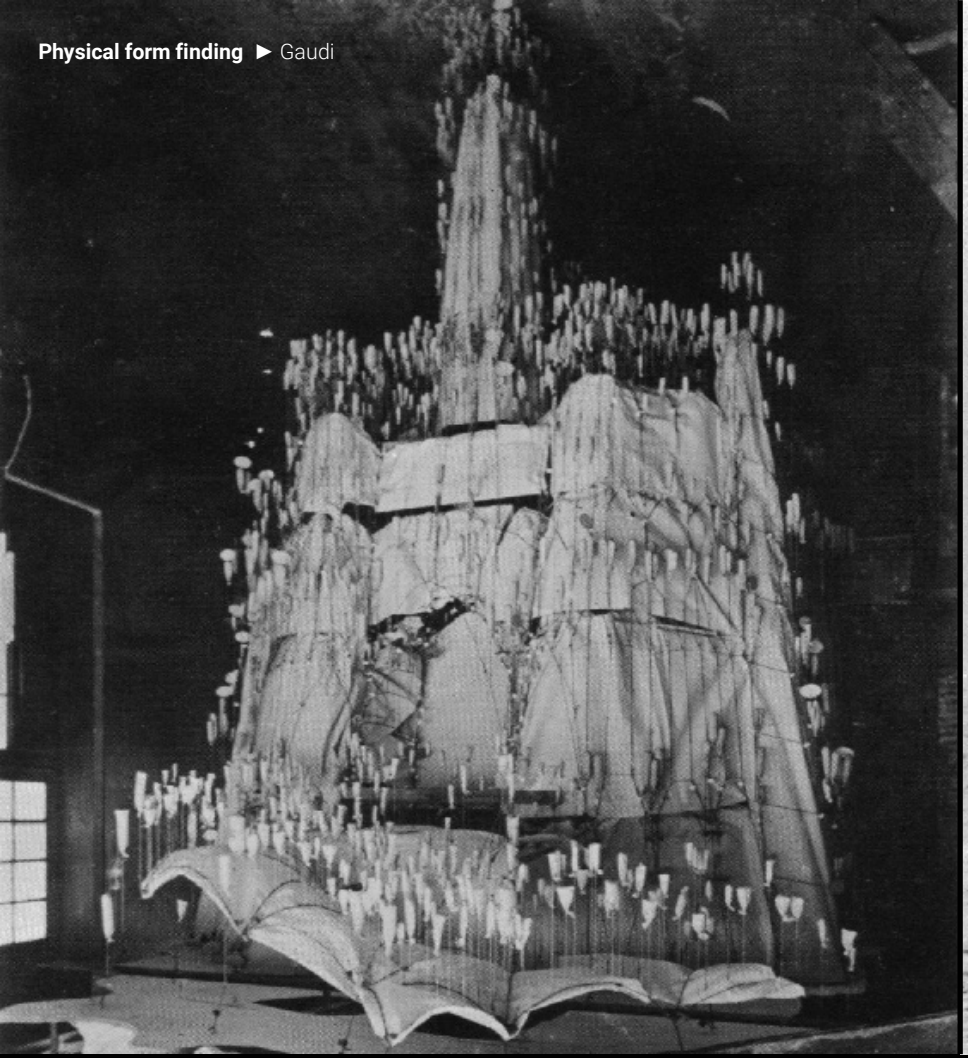
“As hangs the flexible line, so but inverted will stand the rigid arch.”

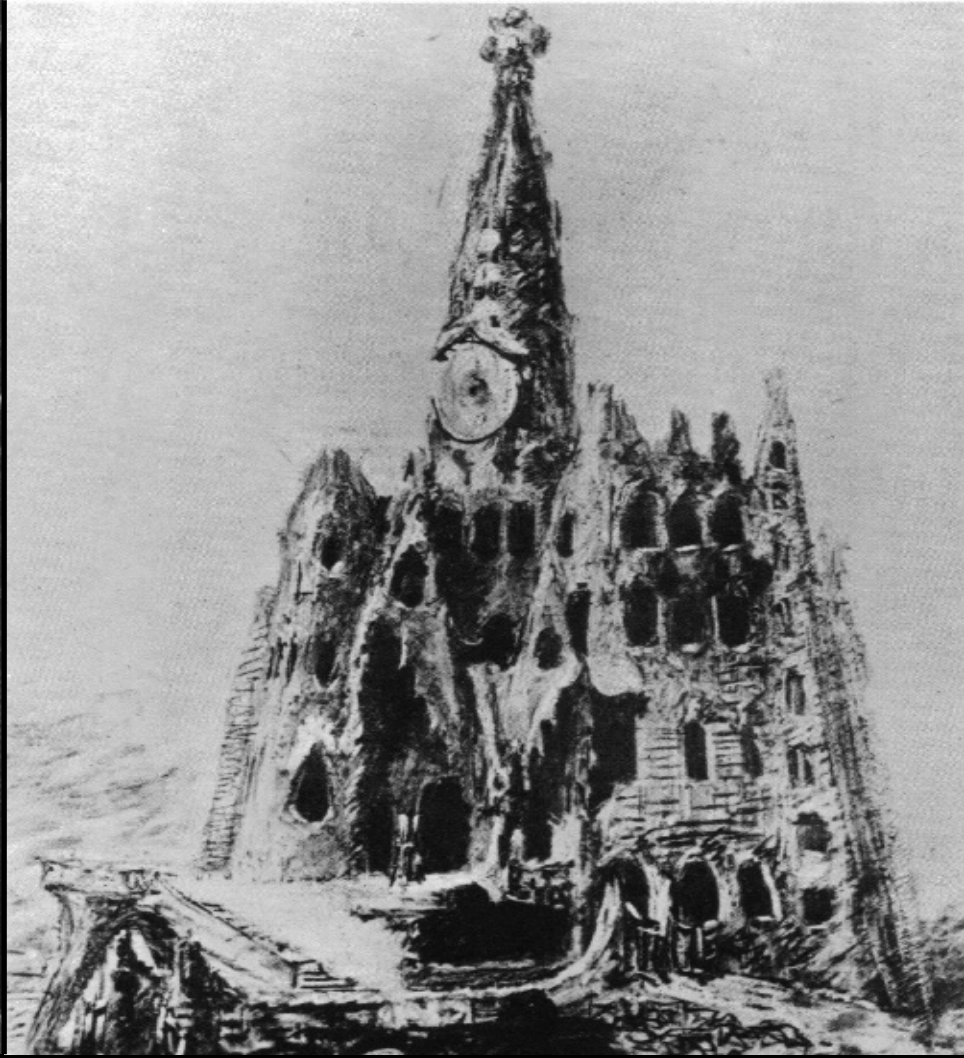
Hooke (1676)

Credit: Prof. G. Wayne Brodland, University of Waterloo



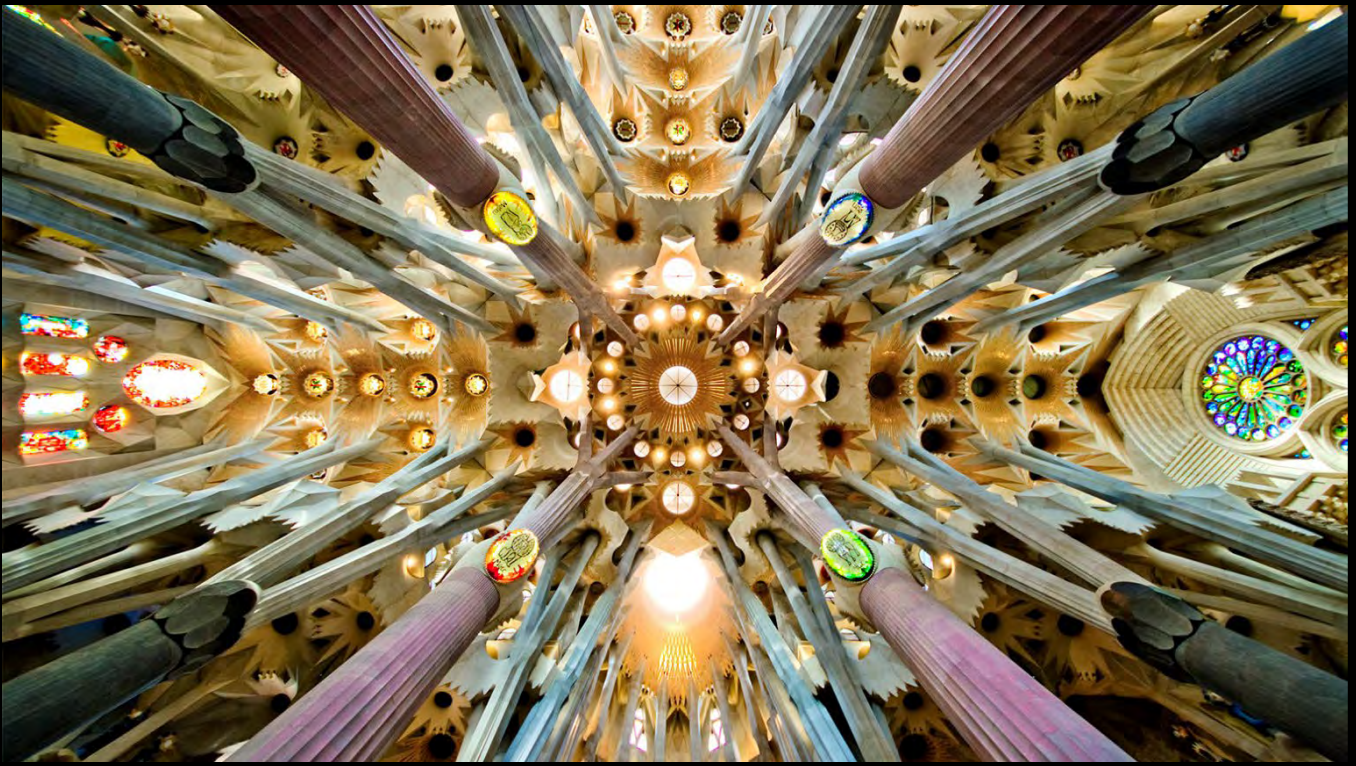
Redesign of Gaudí's hanging model for the Colonia Güell | Images: Beotis (2007)



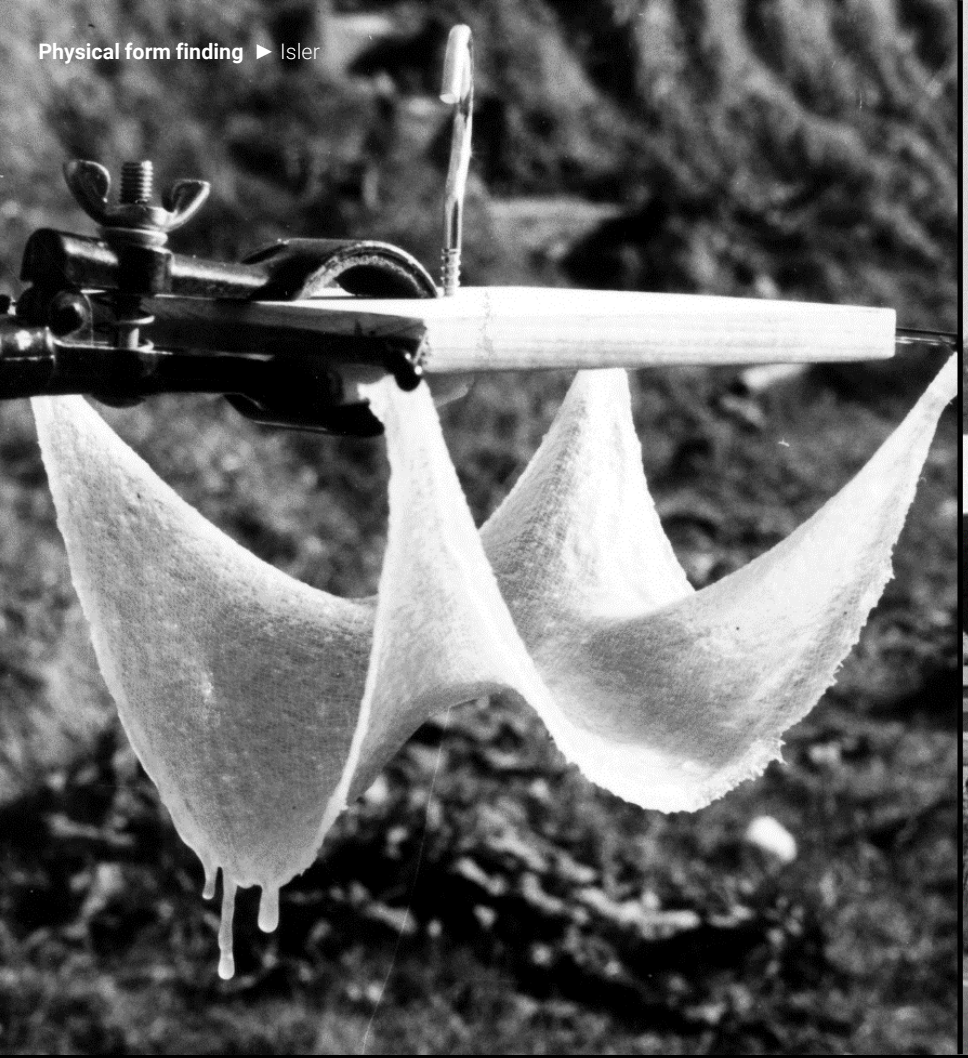


Gaudí's design of its exterior directly sketched on the inverted photograph of the hanging model | Images: Collins, 1963

Gaudí's Church of Colònia Güell, 1898

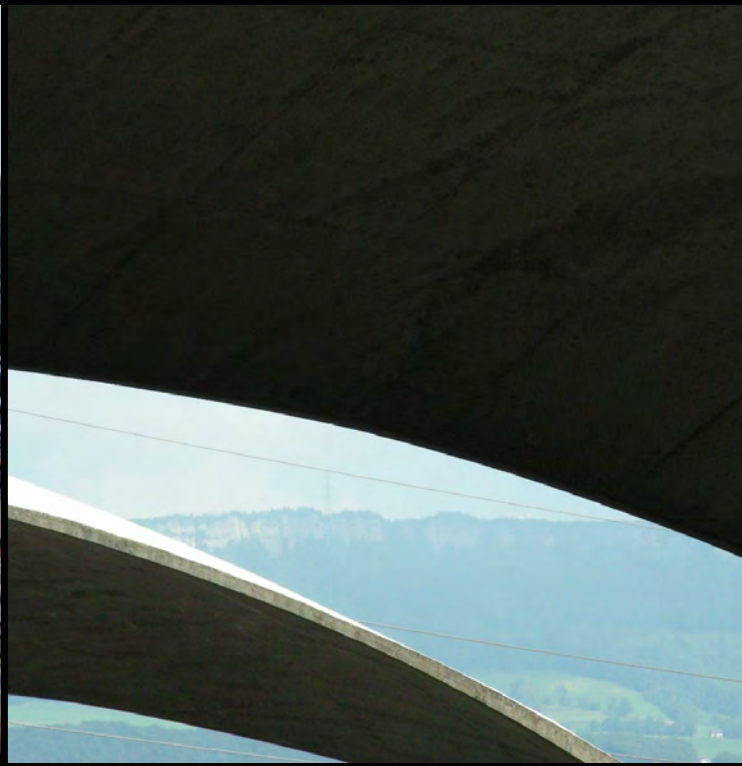


Gaudí's Sagrada Família | Images: Wikipedia



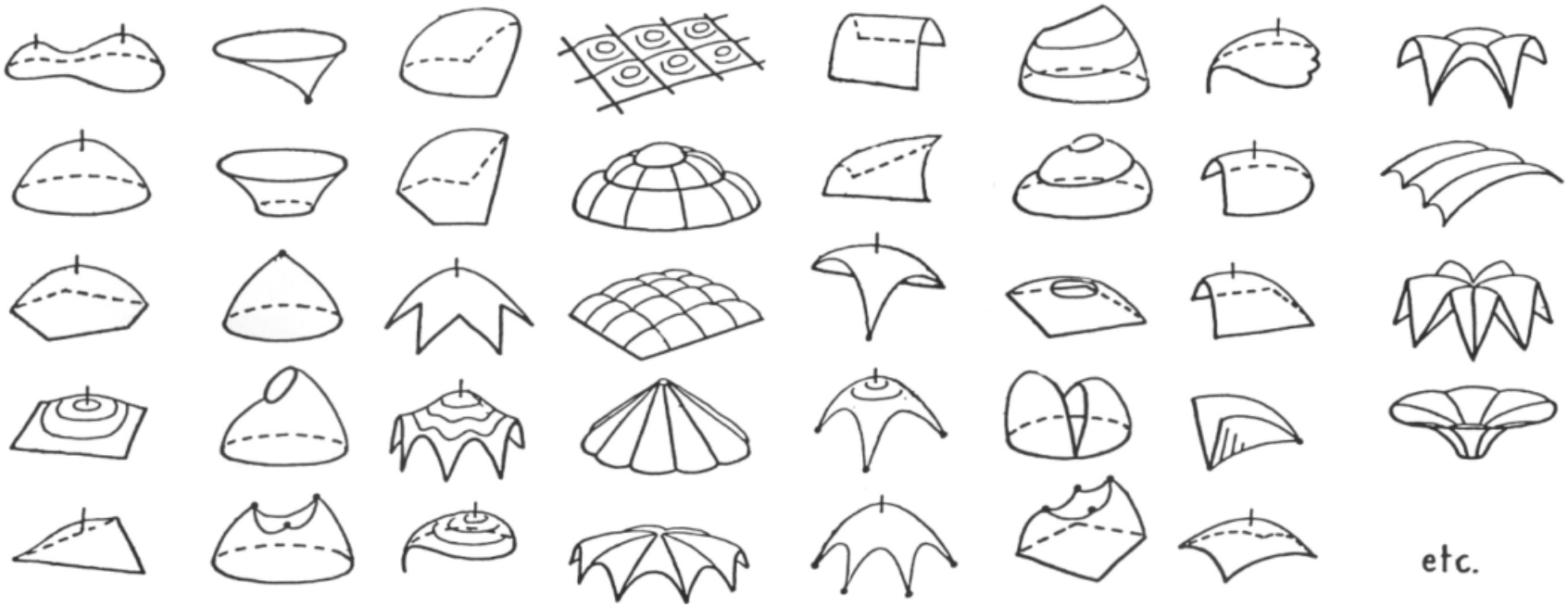
Physical Form Finding Models. | Isler Archive

Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler

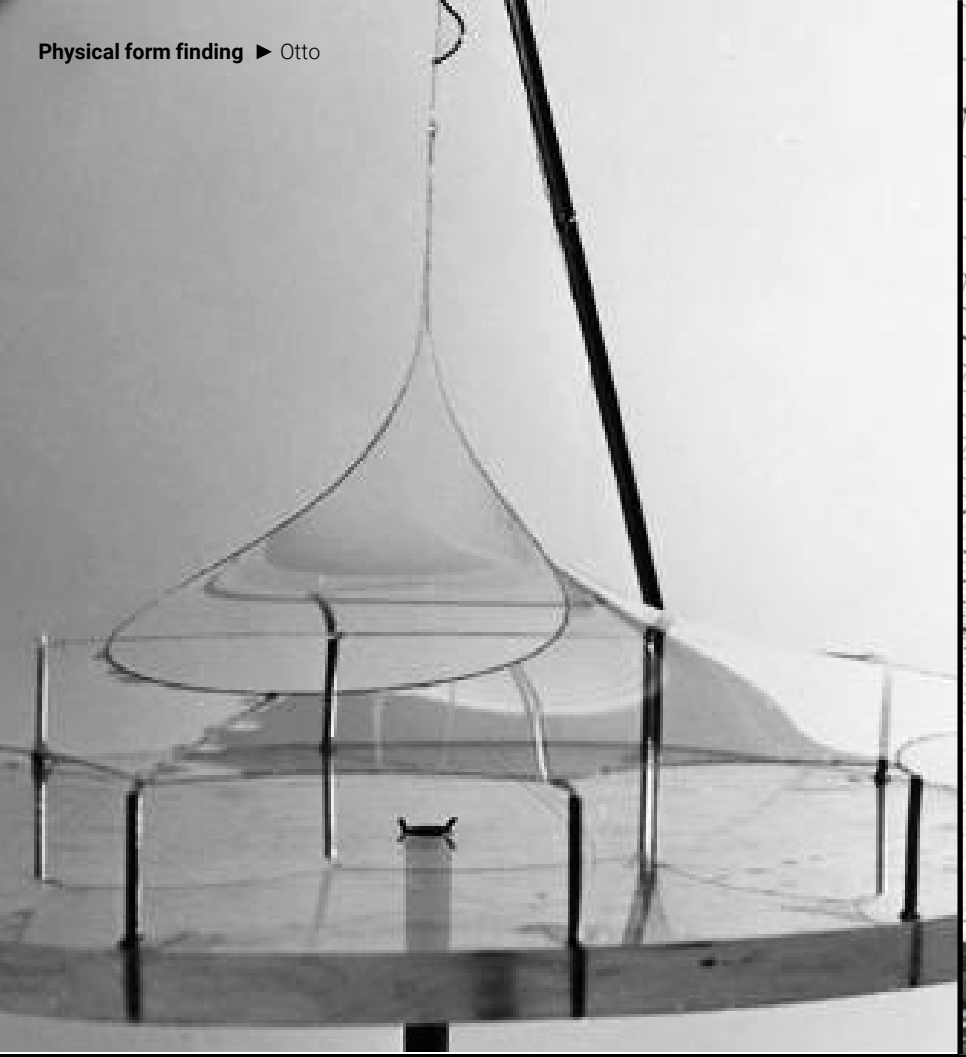


Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler

Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler



"New shapes for shells" Heinz Isler, First Congress of the International Association for Shell Structures (IASS) in 1959



Soap bubble experiments | Frei Otto

German Pavillion at Expo 1967 | Frei Otto

The Institute for Lightweight Structures (IL)
University of Stuttgart

Director: Frei Otto

Roof for Munich Olympic Stadium 1972

Montreal Pavilion measuring model, 1966

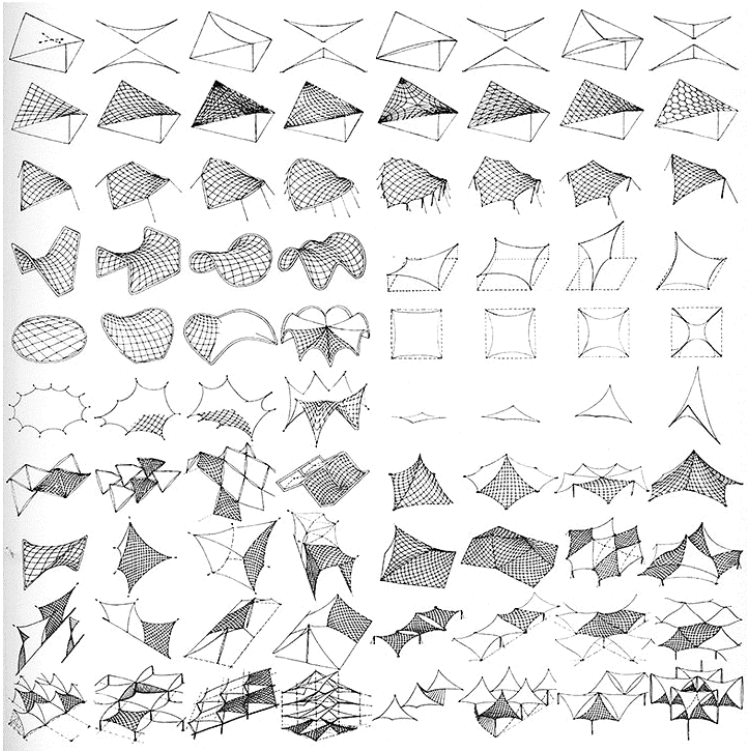
Munich Stadium measuring model, ca. 1970

Computational **form finding**

Form finding

"… forward process in which
parameters are explicitly/directly controlled
to find an 'optimal' geometry of a structure
which is in static equilibrium
with a design loading…"

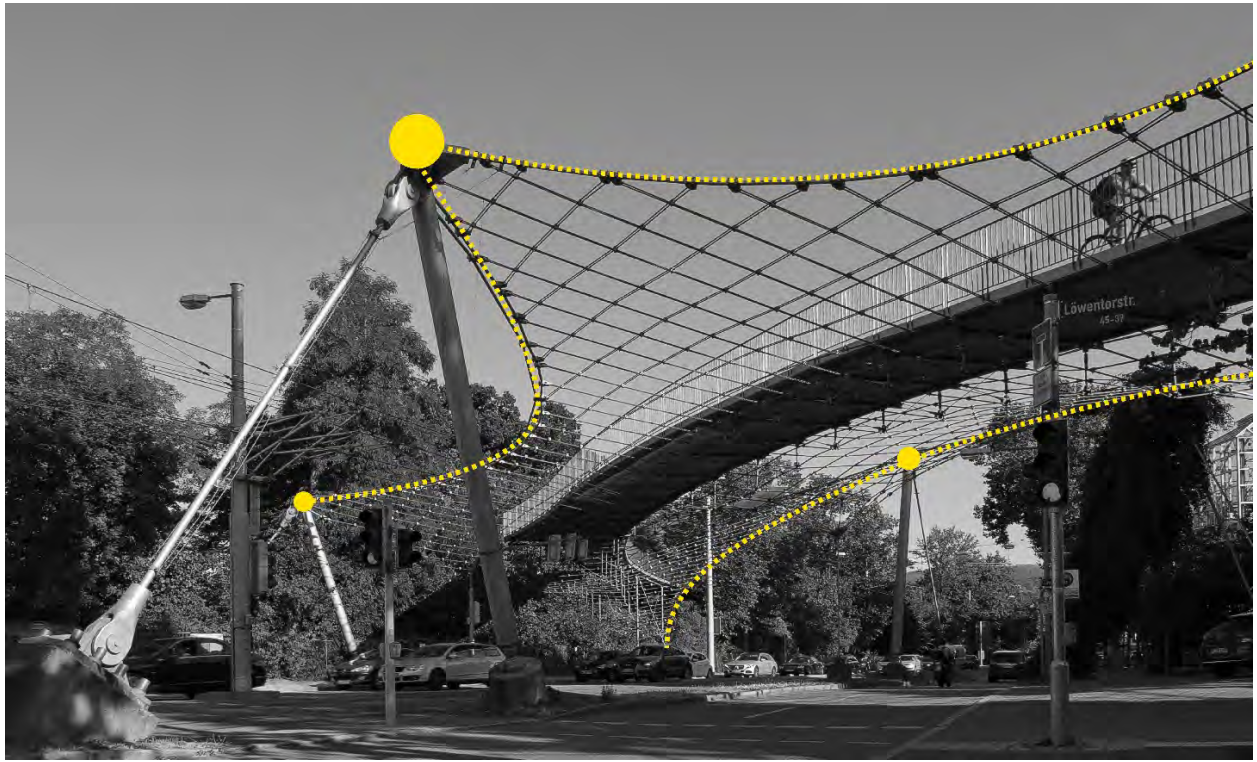
Adriaenssens et al. (2014)



Frei Otto. Casabella 301 1966: 35

Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

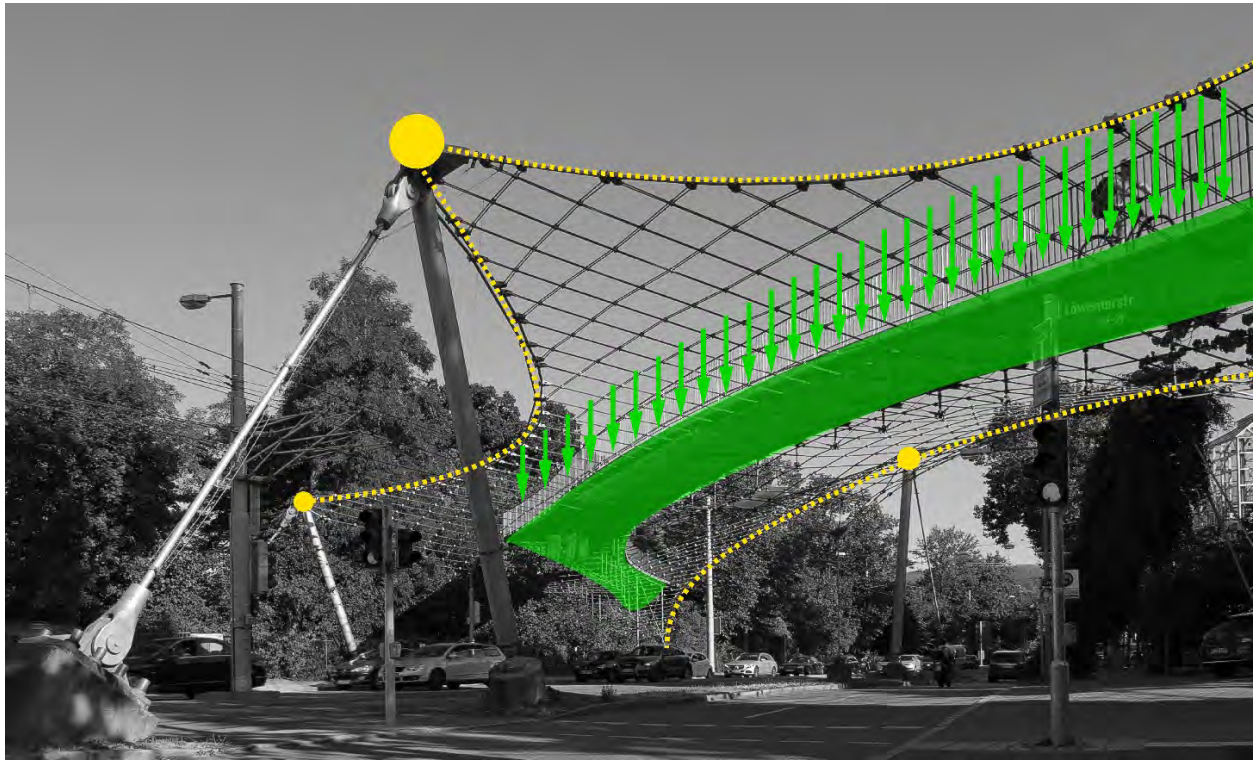
1. Defined boundary conditions



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

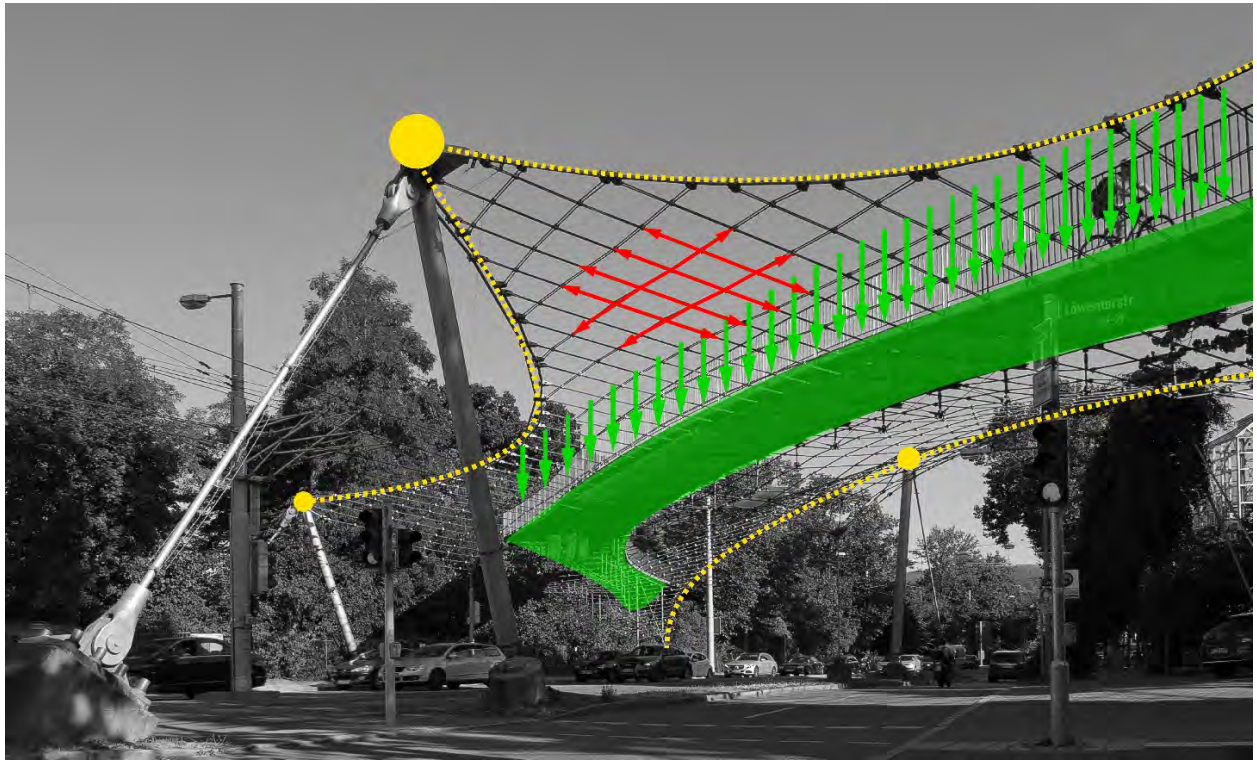
1. Defined boundary conditions

2. Defined loading case



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

1. Defined boundary conditions
2. Defined loading case
3. Defined state of self stress



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

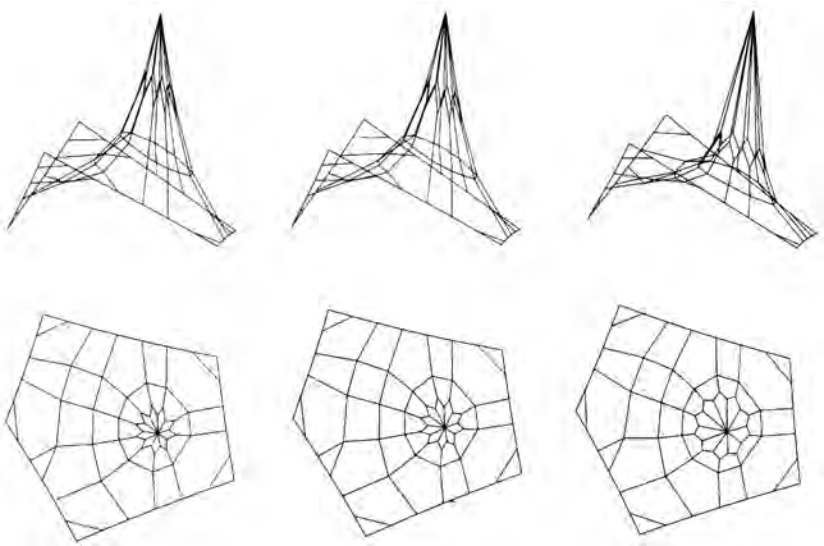


Fig. 7. Global changes in the force densities and lengthening of the radial branches.

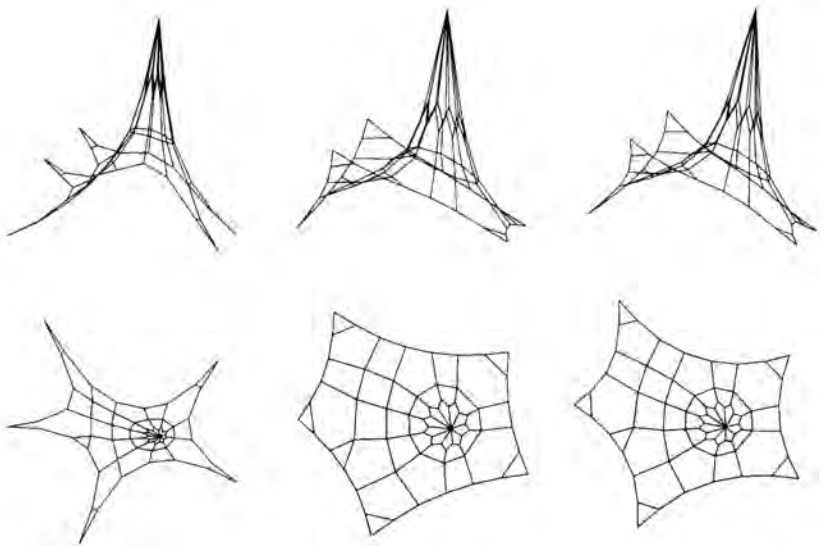


Fig. 8. Global changes in the force densities of the main cables.

"The force density method for form finding and computation of general networks" | Schek (1974)

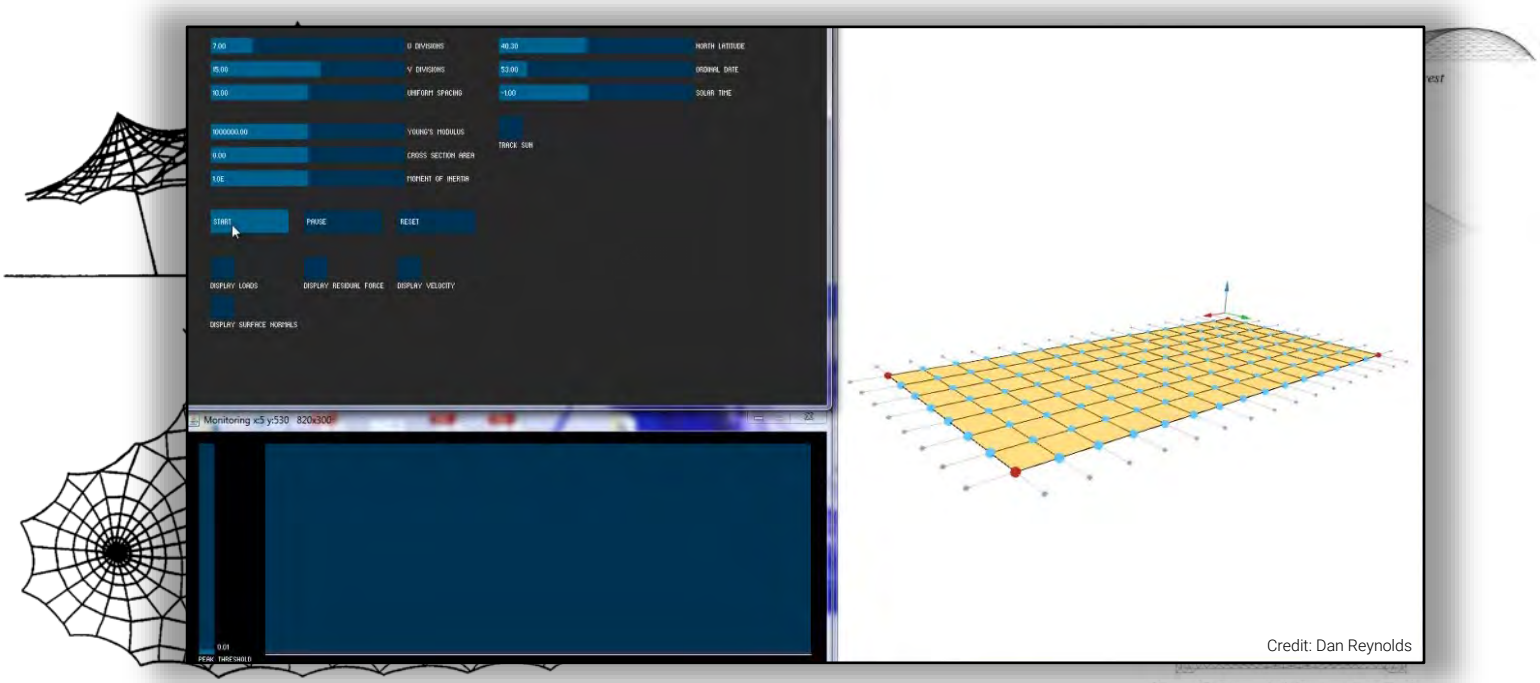


Figure 13. Outwards deflections due to loading

"Form-finding and analysis of prestressed nets and membranes" | Barnes (1988)

"" The analytic and numerical definition of the geometry of the British Museum Great Court Roof" | Williams (2001)

Form finding with graphic statics

JOURNAL OF THE INTERNATIONAL ASSOCIATION FOR SHELL AND SPATIAL STRUCTURES: J. IASS

THRUST NETWORK ANALYSIS: A NEW METHODOLOGY FOR THREE-DIMENSIONAL EQUILIBRIUM

PHILIPPE BLOCK[1] AND JOHN OCHSENDORF[2]

[1]Research Assistant, ph_block@mit.edu, [2]Associate Professor, jao@mit.edu, Building Technology Program, Massachusetts Institute of Technology, Room 5-418, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Editor's Note: The first author of this paper is one of the four winners of the 2007 Hangai Prize, awarded for outstanding papers that are submitted for presentation and publication at the annual IASS Symposium by younger members of the Association (under 30 years old). It is re-published here with permission of the editors of the proceedings of the IASS 2007 Symposium: Structural Architecture: Toward the Future Looking to the Past, held in December 2007 in Venice, Italy.

ABSTRACT

This paper presents a new methodology for generating compression-only vaulted surfaces and networks. The method finds possible funicular solutions under gravitational loading within a defined envelope. Using projective geometry, duality theory and linear optimization, it provides a graphical and intuitive method, adopting the same advantages of techniques such as graphic statics, but offering a viable extension to fully three-dimensional problems. The proposed method is applicable for the analysis of vaulted historical structures, specifically in unreinforced masonry, as well as the design of new vaulted structures. This paper introduces the method and shows examples of applications in both fields.

Keywords: *compression-only structures, unreinforced masonry vaults, funicular analysis, Thrust Network Analysis, reciprocal diagrams, form-finding, lower-bound analysis, structural optimization*

1. INTRODUCTION

Medieval vault builders created complex forms carefully balanced in compression. The structural properties of these sophisticated forms are still poorly understood because of a lack of appropriate analysis methods, i.e. methods relating stability and form. Understanding the mechanics of these vaulted structures leads to new insights for both analysis and design.

Thrust Line Analysis is a powerful graphical method for calculating the range of lower-bound equilibrium solutions of compression-only systems, such as unreinforced masonry structures (Fig. 1). It visualizes the stability of these structures and suggests possible collapse mechanisms [1]. Unfortunately, thrust line analysis is primarily suitable for 2-D cases and this limitation has prevented it from being used for the assessment of complex 3-D structures. While numerical methods

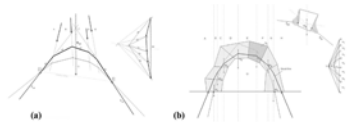


Figure 1. (a) Two possible compression-only equilibrium shapes for a random set of loads, and (b) an interactive thrust-line application developed in [1]: the user can adopt the geometry by dragging control points and the structural feedback, in the form of a thrust-line, is updated in real-time. The magnitudes of the forces in the system are visualized in the accompanying funicular polygon (right).

Vol. 48 (2007) No. 3 December n. 155

based on elastic solutions give one possible answer, they no longer suggest better form as was inherent to the more holistic graphical methods.

There is a real need for tools to better understand and visualize the stability of compression-only structures, such as historic unreinforced masonry structures, as well as design tools that suggest better form. Both problems are related to finding axial force structures in equilibrium acting only in compression or tension. Currently, graphic statics provides a holistic analysis and design tool for two-dimensional structures. With today's availability of powerful virtual 3-D and parametric environments, the following question arises: can a fully three-dimensional version of thrust-line analysis provide the same freedom to explore the infinite equilibrium solutions for a certain loading condition?

2. METHODOLOGY

The *Thrust-Network Analysis* method presented in this paper is inspired by O'Dwyer's work on funicular analysis of vaulted masonry structures [2]. It is extended by adding the concept of duality between geometry and the in-plane internal forces of networks [3].

2.1. Reciprocal figures

The duality between the geometry of a network and its internal forces is an old concept, first explained by Maxwell [4]. He called this relationship *reciprocal* and defined it as follows: *"Two plane figures are reciprocal when they consist of an equal number of lines, so that corresponding lines in the*

Figure 2. The primal grid "" and dual grid " are related by a reciprocal relationship. The equilibrium of a node in one of them is guaranteed by a closed polygon in the other and vice versa. The labeling uses Bow's notation [6].*

two figures are parallel, and corresponding lines which converge to a point in one figure form a closed polygon in the other." This means that the equilibrium of a node in the first diagram is represented by a closed polygon in the second diagram and vice versa (Fig. 2). Graphic statics is based on this principle [5].

2.2. Assumptions

The proposed method produces funicular (compression-only) solutions for loading conditions where all loads are applied in the same direction, as is the case for gravitational loading. Since the solutions are compression-only this also means that the vaults can never curl back onto themselves, which would demand some elements to go into tension. The resulting three-dimensional networks can represent load paths throughout a structure. There is no constraint on the length of the branches

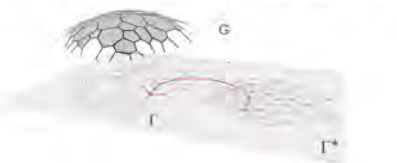


Figure 3. Relationship between compression shell (G), its planar projection (grid Γ) and the reciprocal diagram, dual grid (Γ) to determine equilibrium.*

JOURNAL OF THE INTERNATIONAL ASSOCIATION FOR SHELL AND SPATIAL STRUCTURES: J. IASS

Figure 7. (a) Possible thrust values for this given vault range from 21% to 32% of its total weight. (b) 3-D web and rib action with the forces mainly spanning between the ribs as represented in the dual grid (c).

realm of lower-bound analysis. Put simply, if a compression-only network can be found that fits within the boundaries of a vault, then the vault will stand in compression. Note that if we want a solution without any tension (hinges) in the section, we want the solution to lie within the middle third of the section (defined by equation 6). This is a powerful concept for understanding the stability and proximity to collapse of such structures. Additional reading on this topic can be found in Heyman [10], O'Dwyer [2], Boothby [11] and Block et al. [1].

The method is particularly appropriate for historic masonry vaults because their self-weight is the dominant load. The range of possible equilibrium states, bounded by a minimum and maximum thrust, can be produced (Fig. 7a). We are interested in this range of thrust values because it provides the most useful characterization of the structural behavior of the vault. The minimum (or passive) thrust state represents the least amount this vault will push sideways onto its neighboring elements, as a function of its self-weight and shape. The maximum (or active) state of thrust on the other hand represents the largest horizontal force this vault can provide. So, this value demonstrates how much horizontal force this vault can safely take from its neighboring elements.

Figure 7b shows a solution with three-dimensional web action. The distribution of the horizontal components in the network is represented in its dual grid (Fig. 7c). More detailed applications of this methodology to masonry vaults can be found in [12].

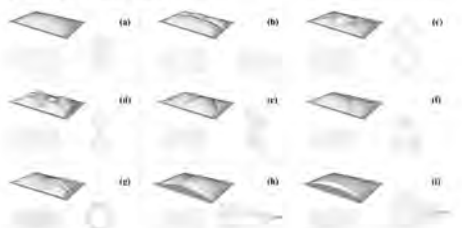
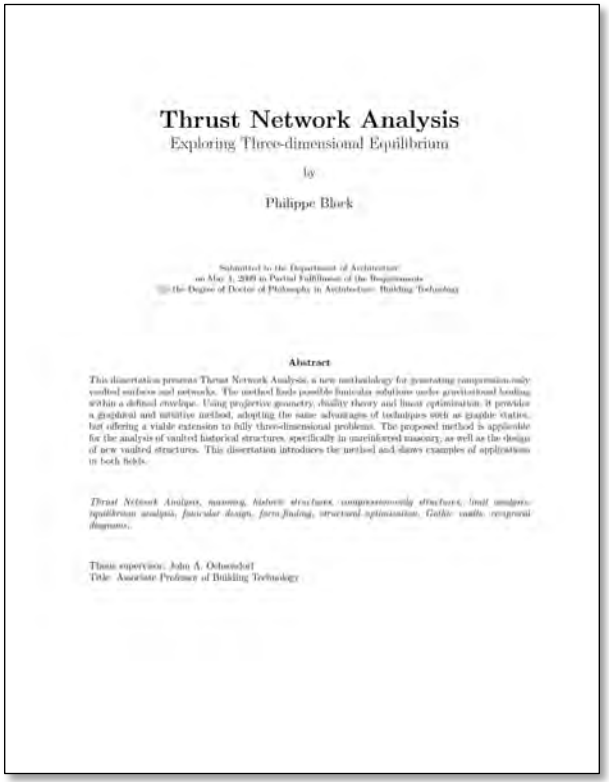


Figure 8. A series of examples, starting from a regular rectangular grid, showing the relationship between the primal and dual grid and the corresponding solutions.



Thrust Network Analysis

Exploring Three-dimensional Equilibrium

by

Philippe Block

Submitted to the Department of Architecture
on May 1, 2009 in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Architecture: Building Technology

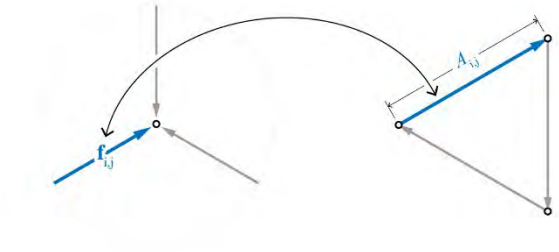
Abstract

This dissertation presents Thrust Network Analysis, a new methodology for generating compression-only vaulted surfaces and networks. The method finds possible funicular solutions under gravitational loading within a defined envelope. Using projective geometry, duality theory and linear optimization, it provides a graphical and intuitive method, adopting the same advantages of techniques such as graphic statics, but offering a viable extension to fully three-dimensional problems. The proposed method is applicable for the analysis of vaulted historical structures, specifically in unreinforced masonry, as well as the design of new vaulted structures. This dissertation introduces the method and shows examples of applications in both fields.

Thrust Network Analysis, masonry, historic structures, compression-only structures, limit analysis, equilibrium analysis, funicular design, form finding, structural optimization, Gothic vaults, reciprocal diagrams.

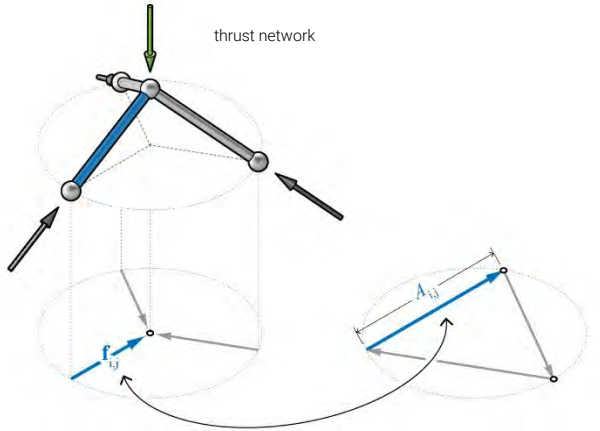
Thesis supervisor: John A. Ochsendorf
Title: Associate Professor of Building Technology

"Thrust Network Analysis" | Block (2009)



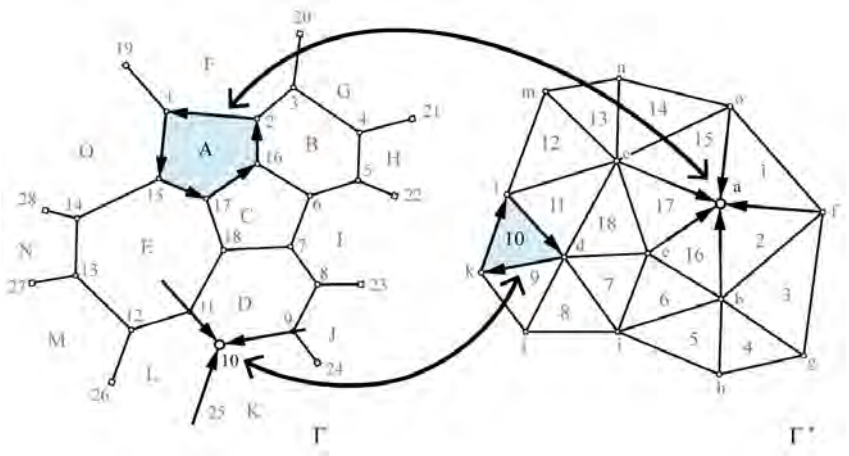
form diagram force diagram

2 D

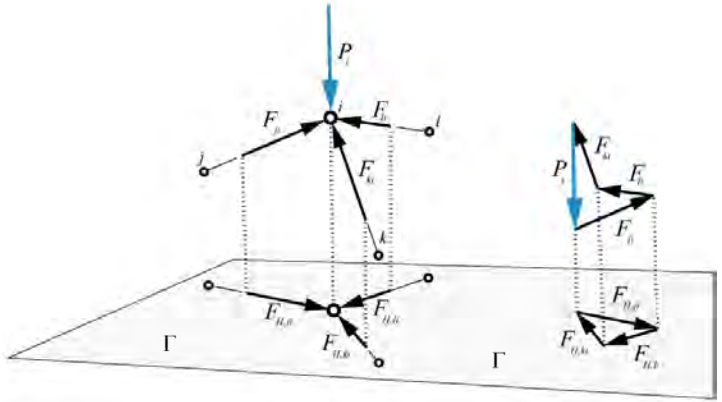


form diagram force diagram

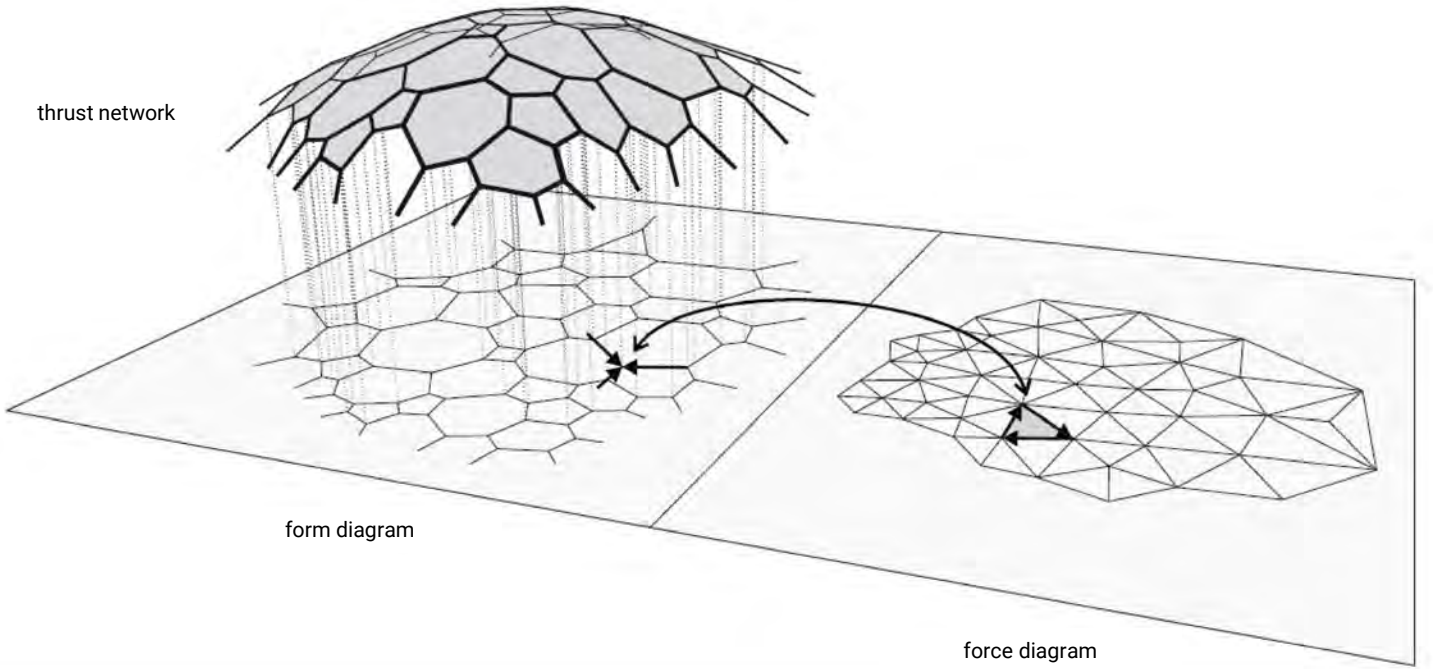
2.5 D



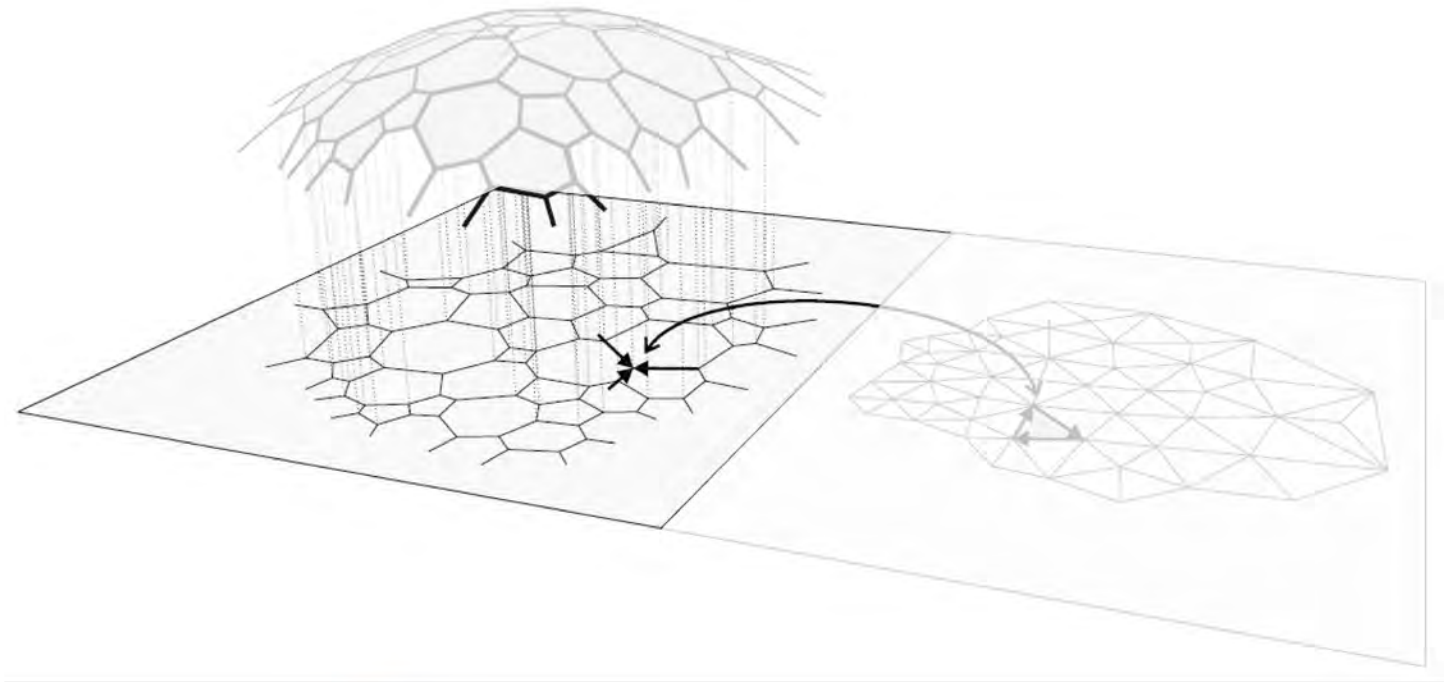
Form Diagram



Force Diagram

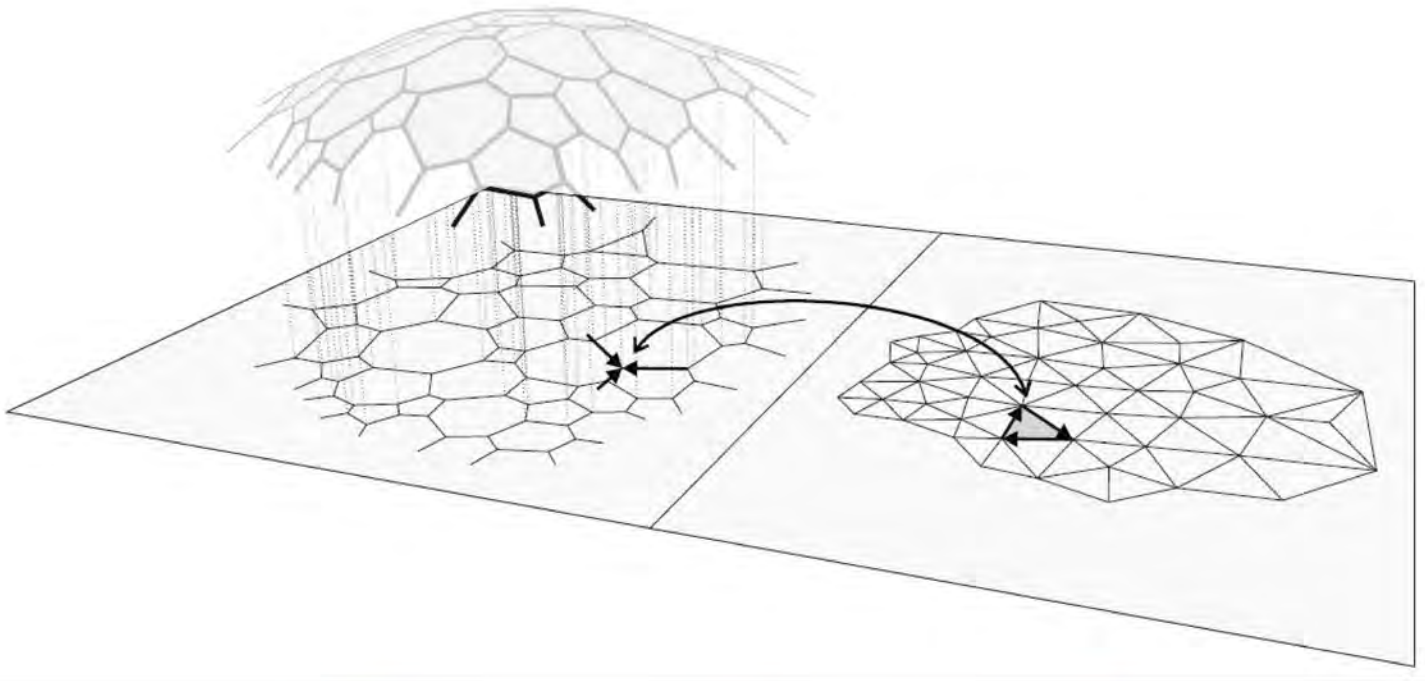


"Thrust Network Analysis" | Block (2007)



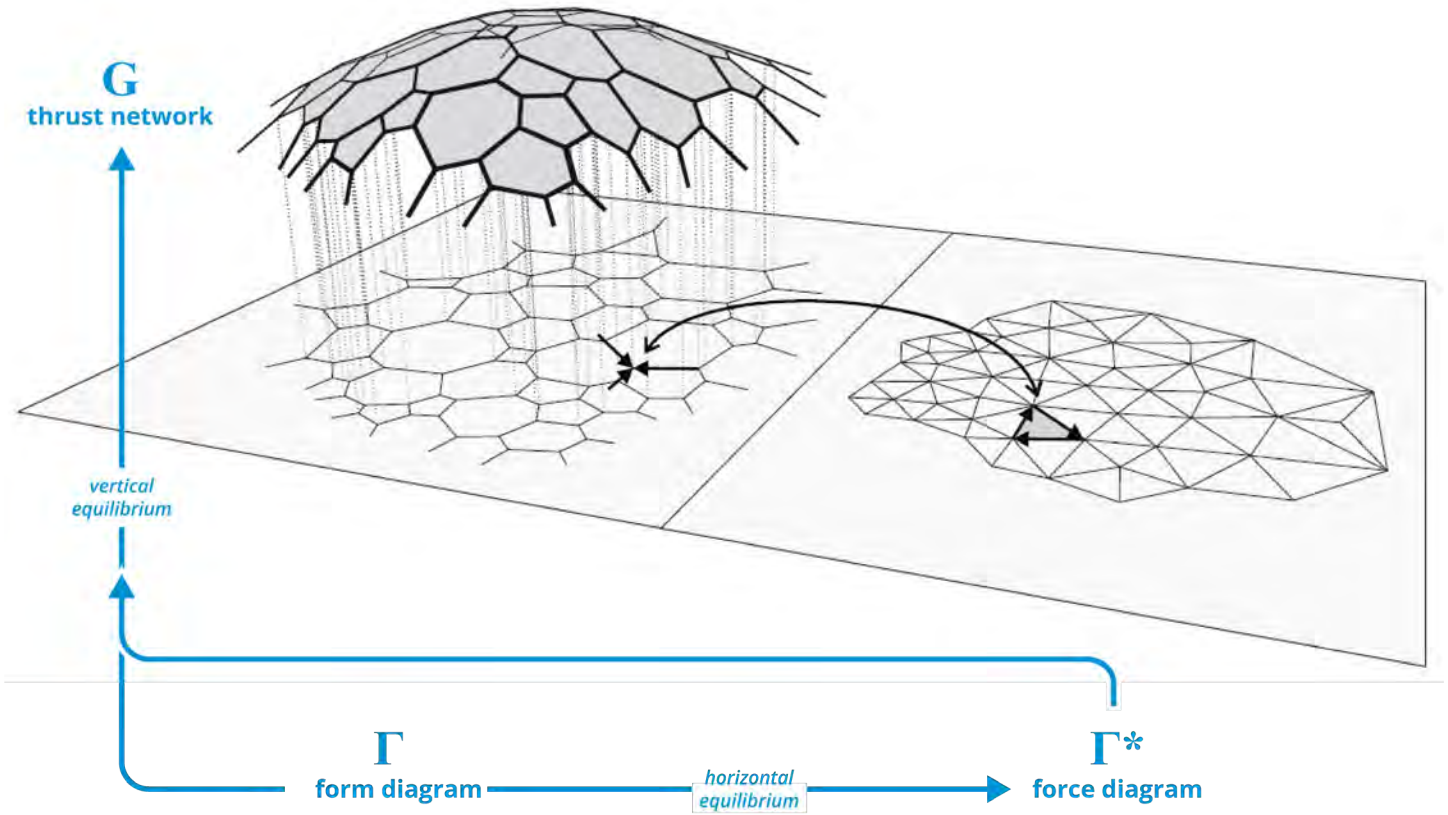
Γ
form diagram

"Thrust Network Analysis" | Block (2007)



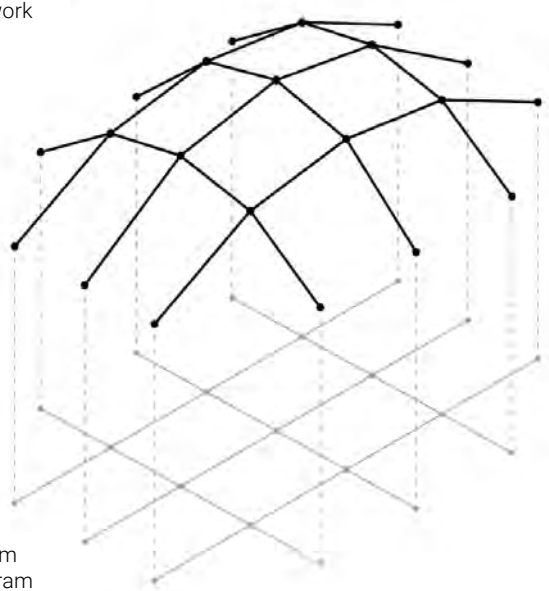
Γ form diagram —*horizontal equilibrium*→ Γ^* force diagram

"Thrust Network Analysis" | Block (2007)

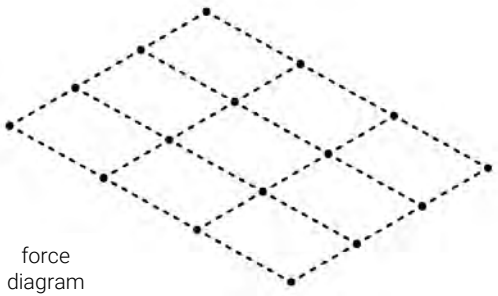


"Thrust Network Analysis" | Block (2007)

thrust network



form diagram



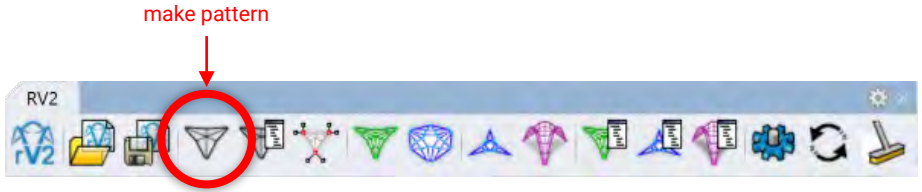
force diagram

"Funicular Shell Design" | Rippmann (2018)

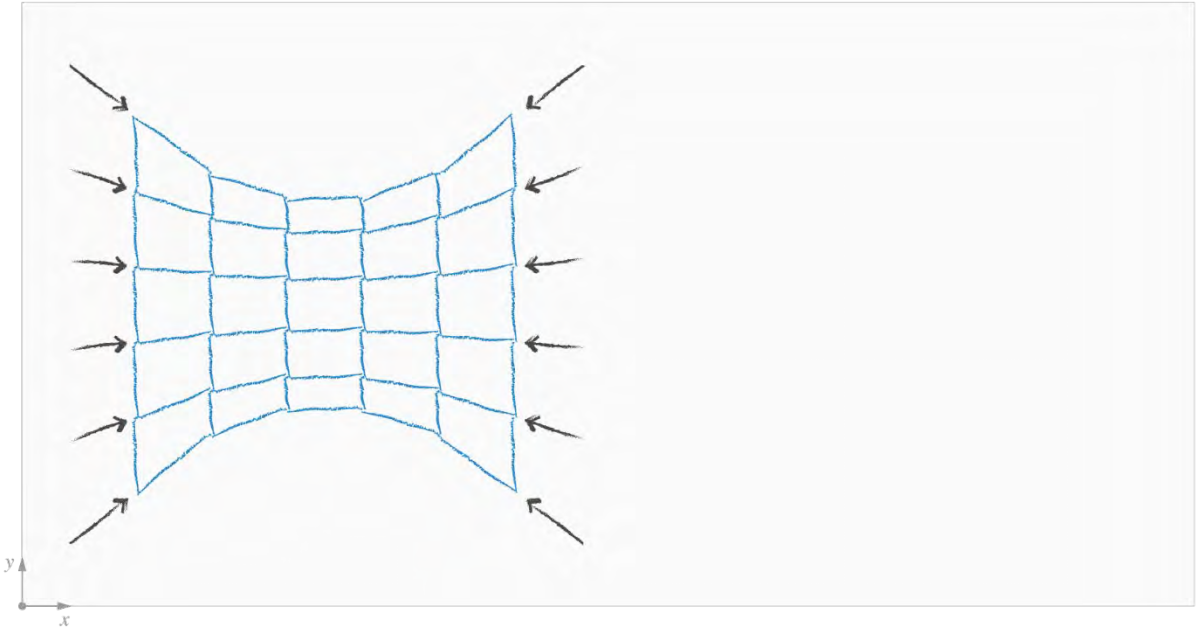
RV2 workflow

TNA steps

1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



1. Pattern

2. Boundary conditions

3. Form diagram

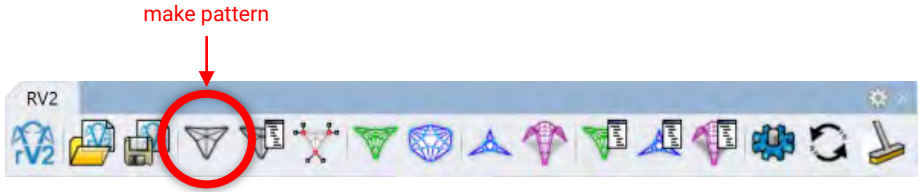
4. Dual diagram

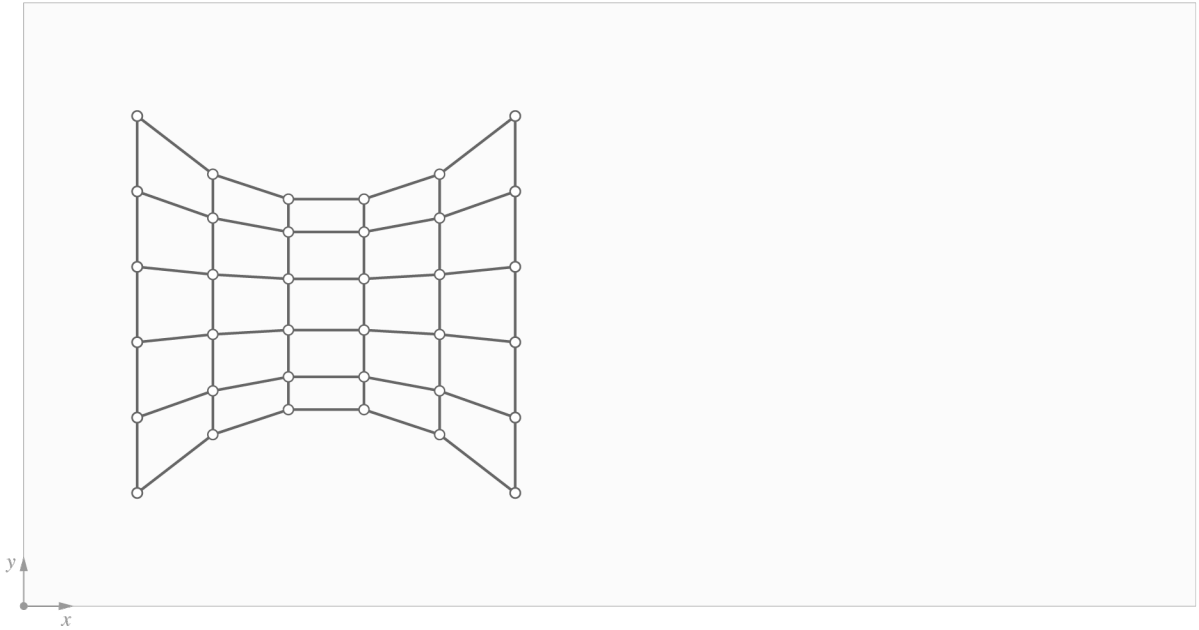
5. Horizontal equilibrium

6. Force diagram

7. Vertical equilibrium

8. Thrust diagram





1. Pattern

2. Boundary conditions

3. Form diagram

4. Dual diagram

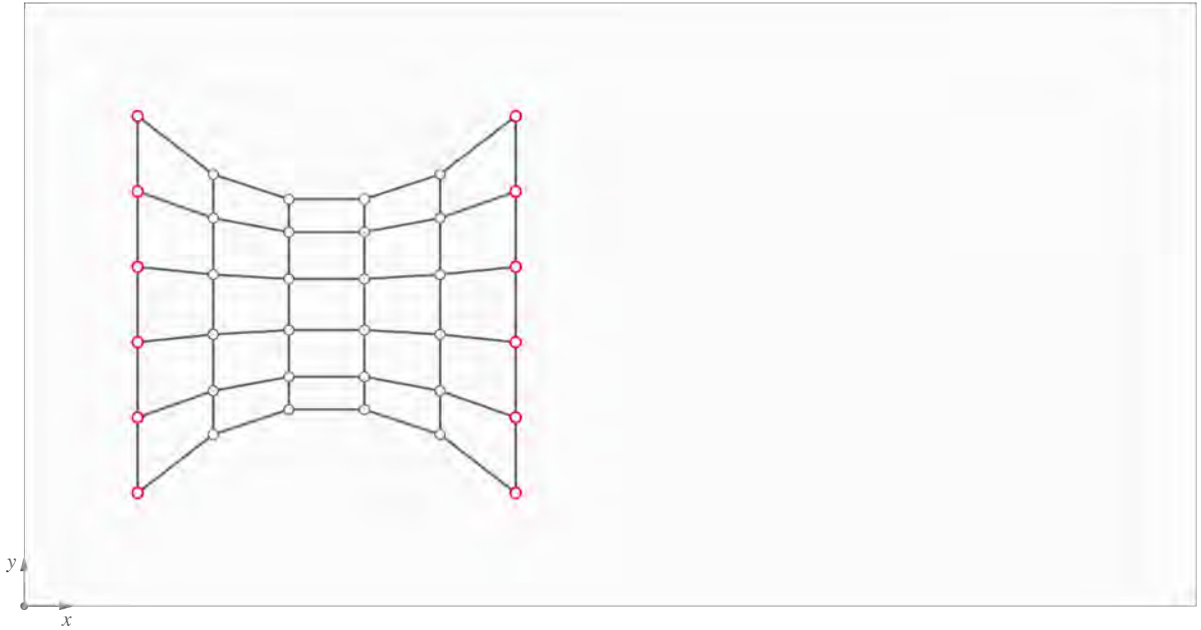
5. Horizontal equilibrium

6. Force diagram

7. Vertical equilibrium

8. Thrust diagram

define boundary conditions



1. Pattern

2. Boundary conditions

3. Form diagram

4. Dual diagram

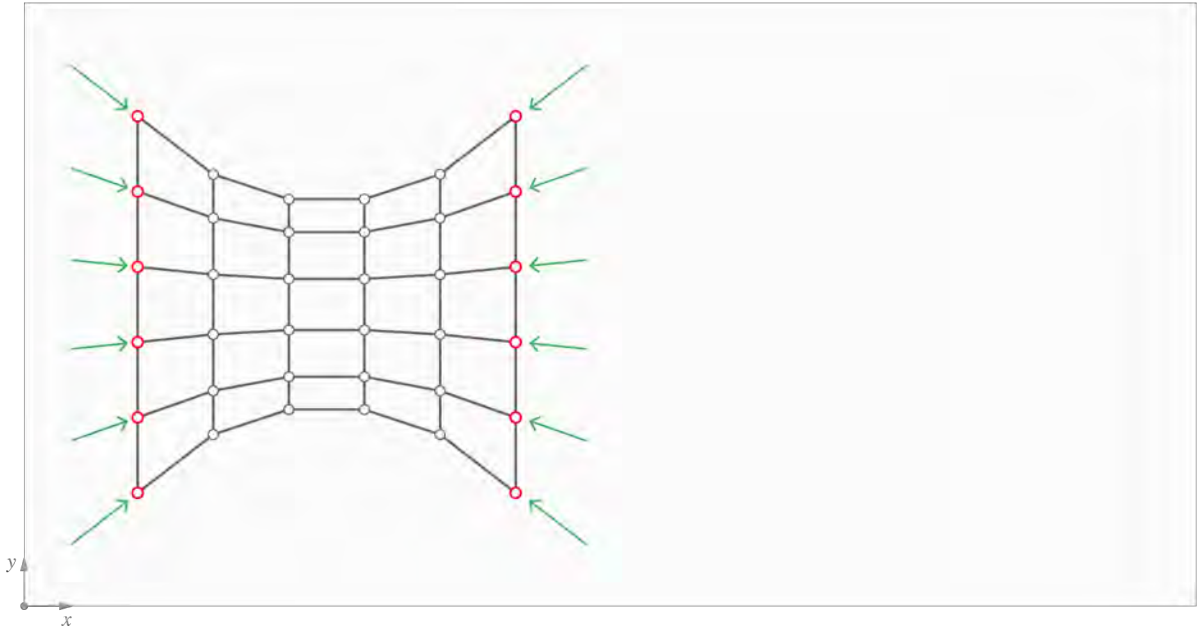
5. Horizontal equilibrium

6. Force diagram

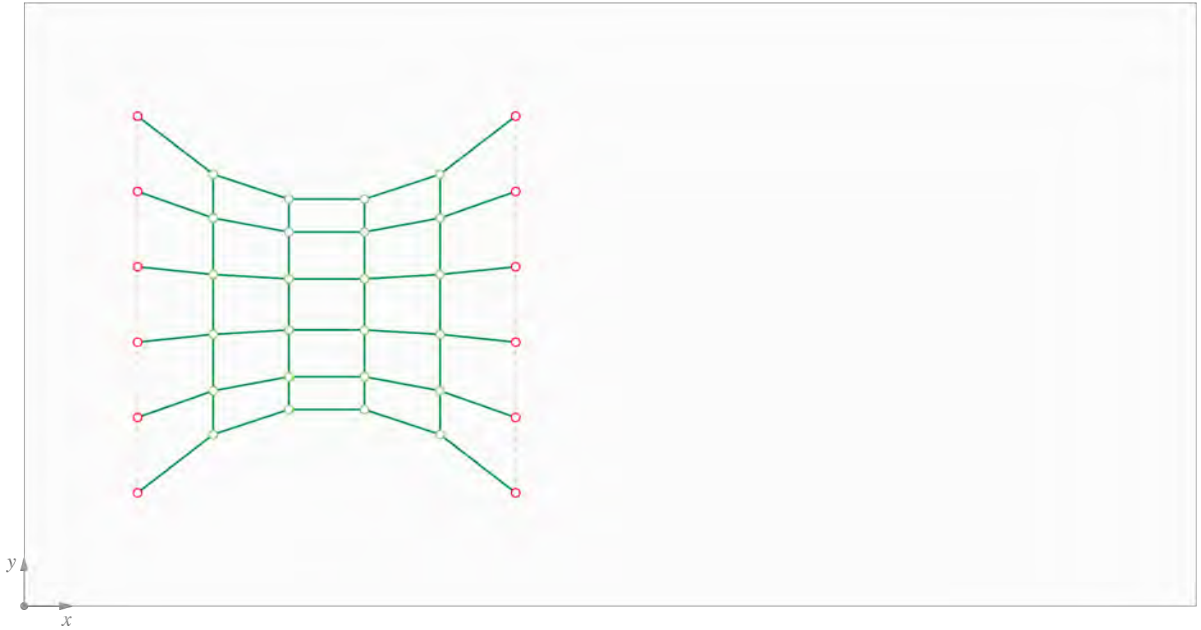
7. Vertical equilibrium

8. Thrust diagram

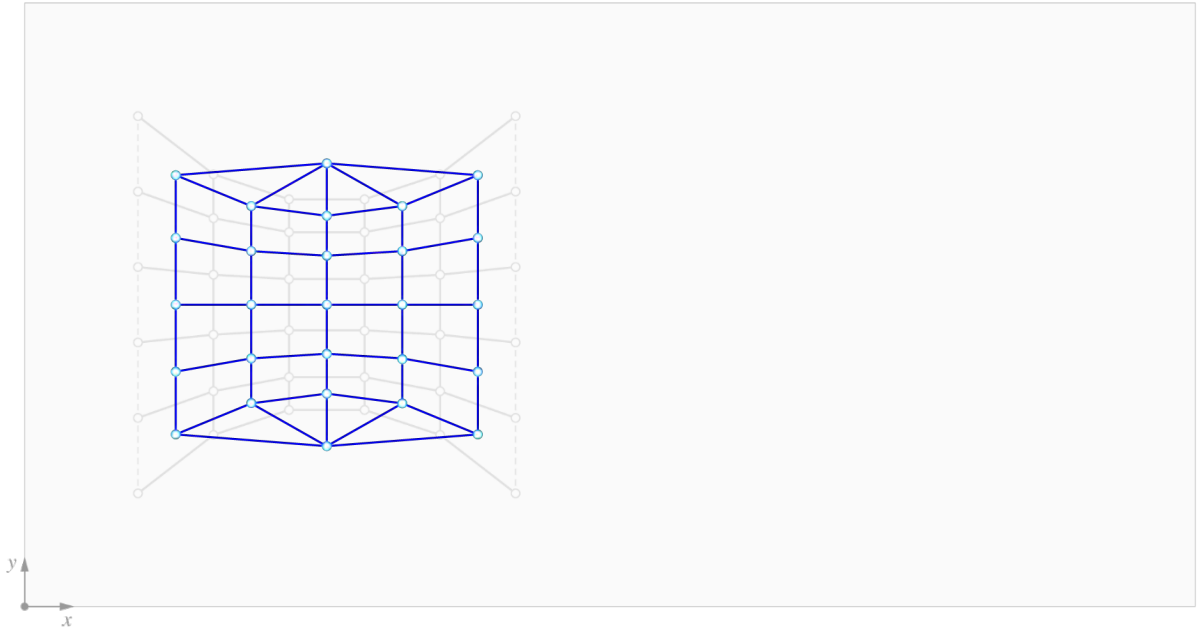
define boundary conditions



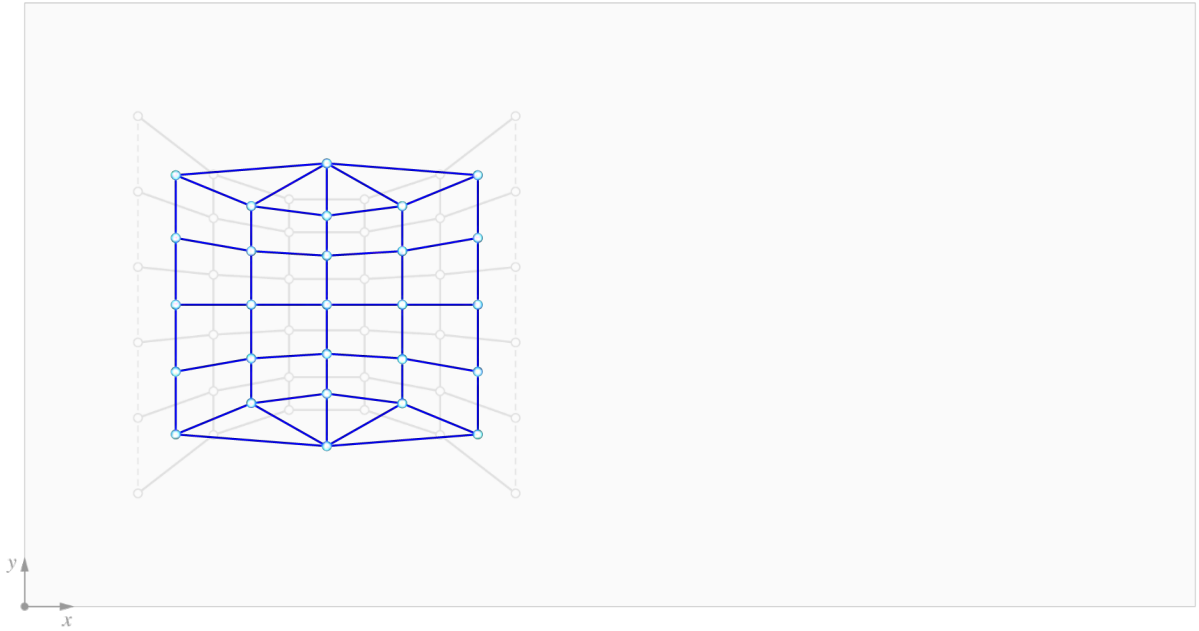
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

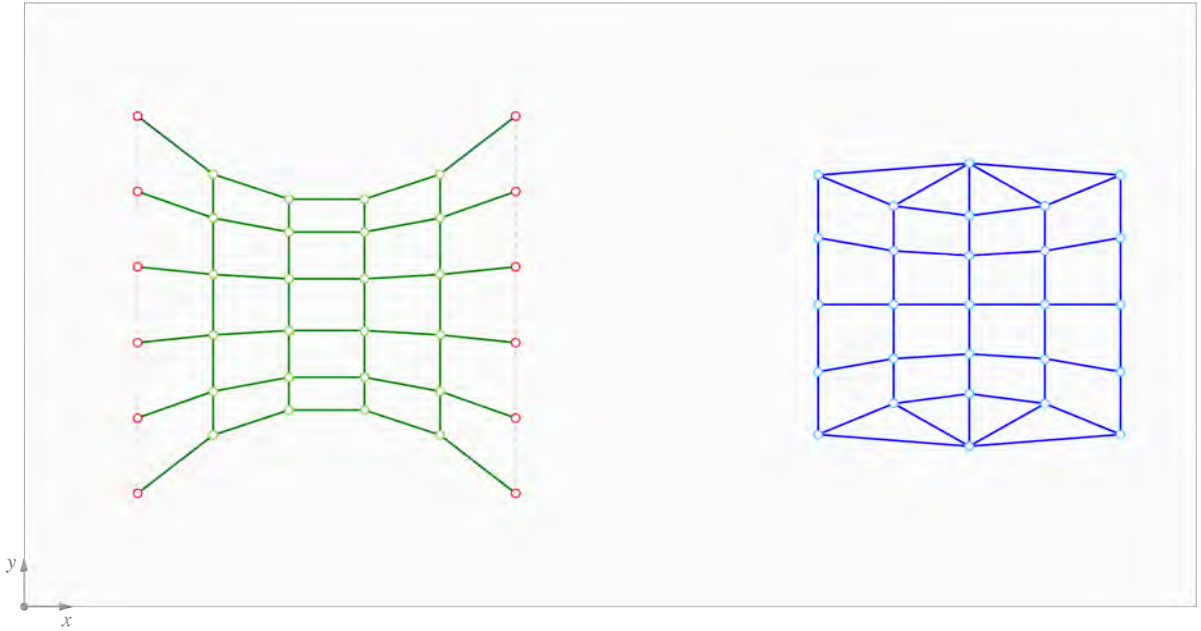


1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



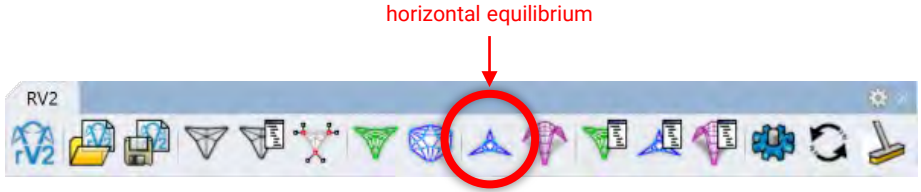
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

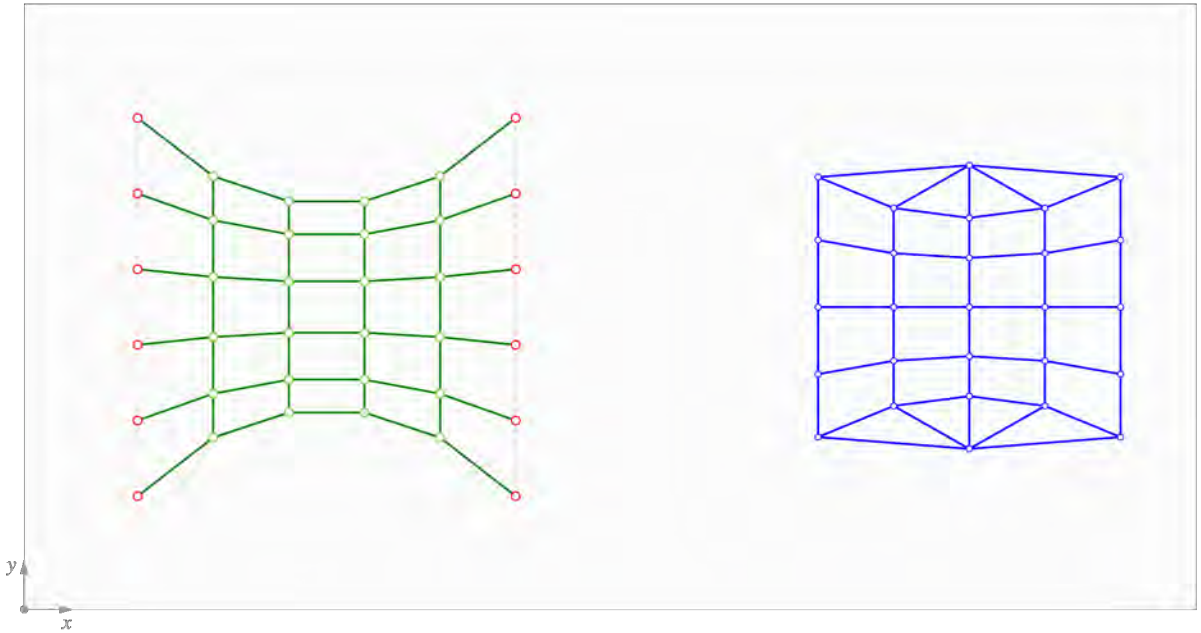




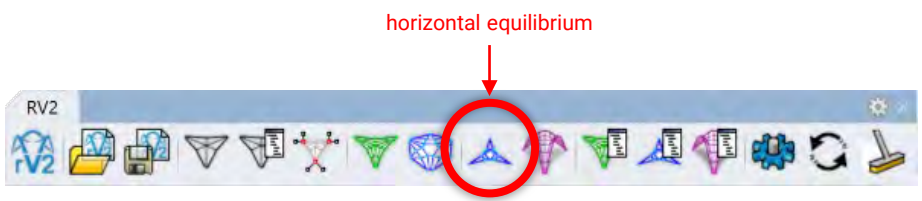
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

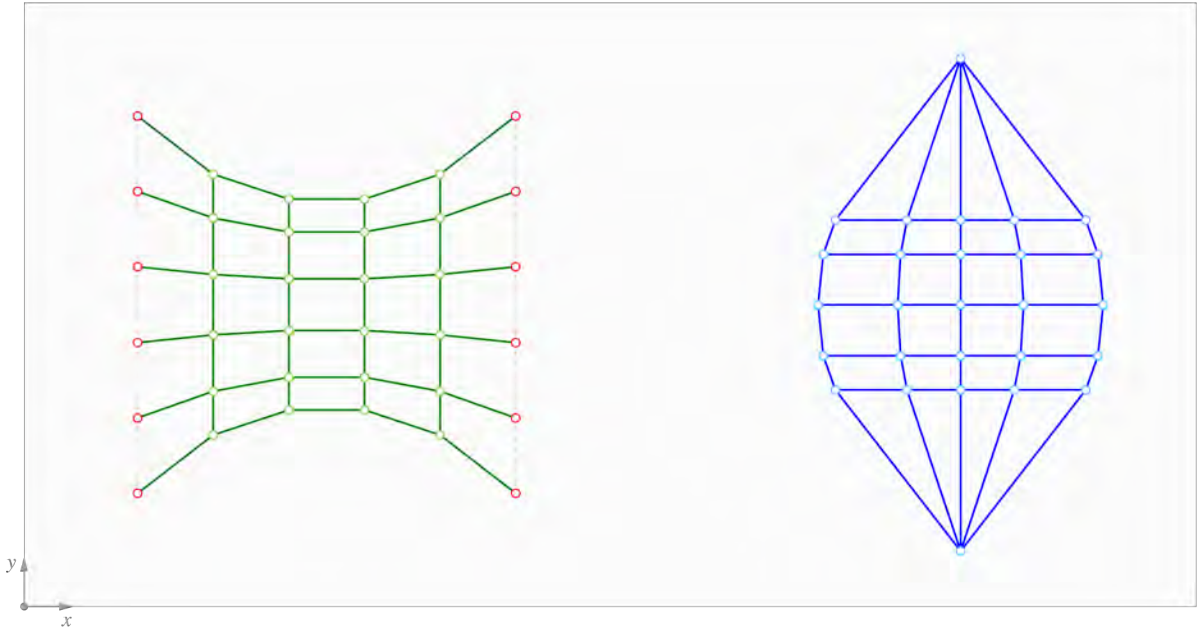
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

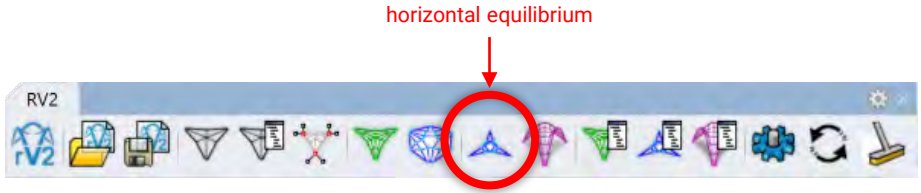


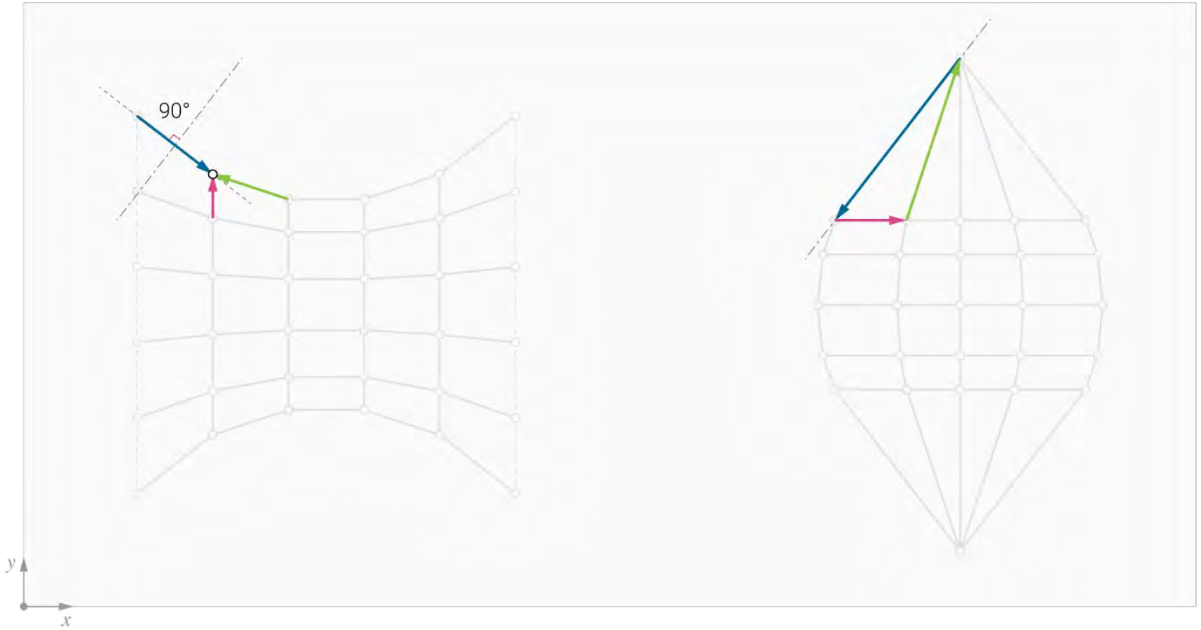


1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

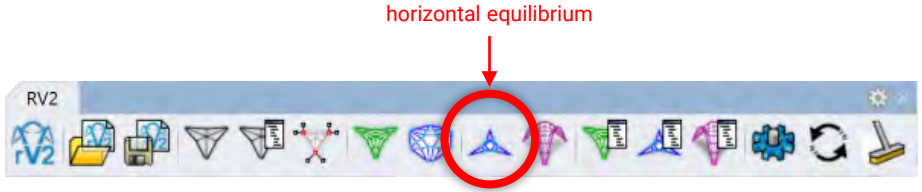


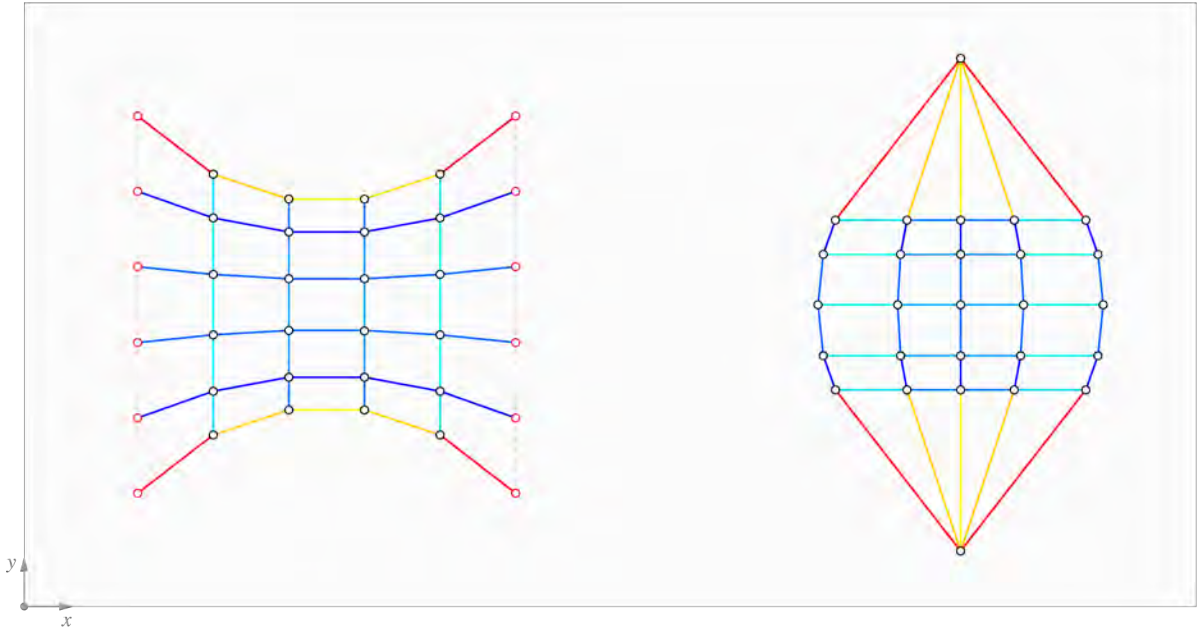




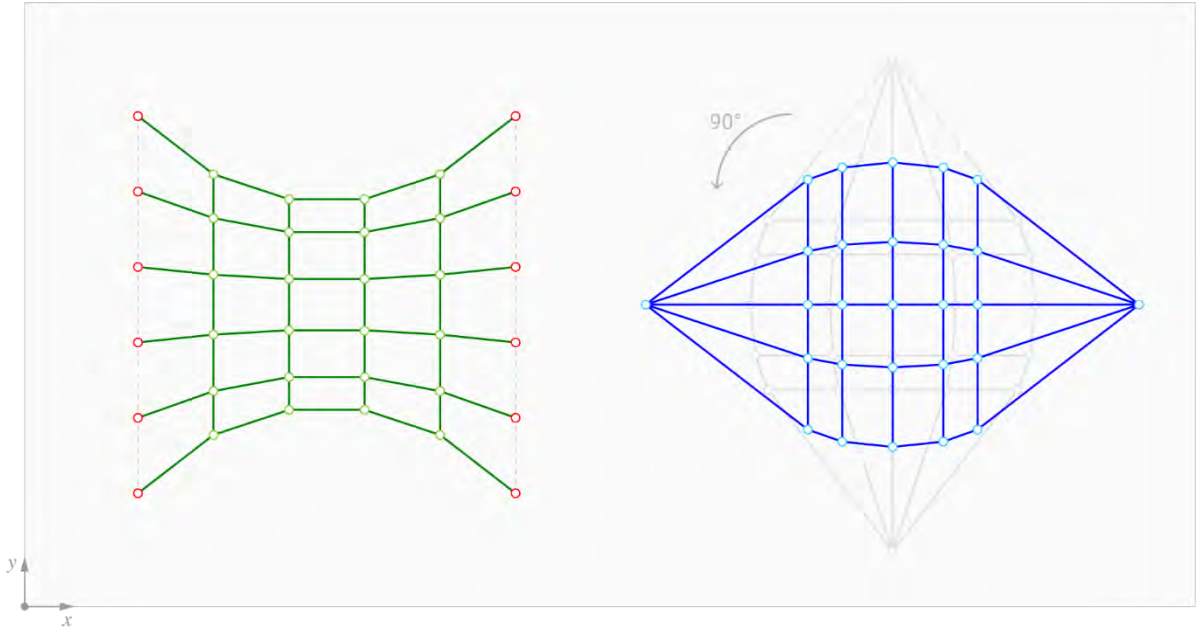


1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

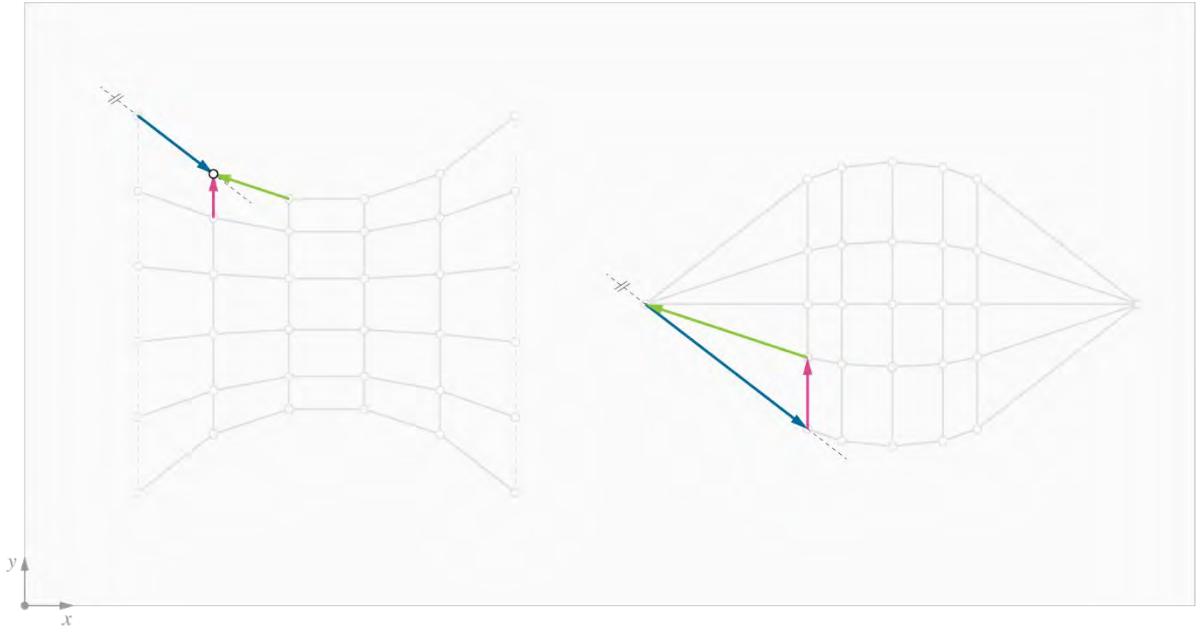




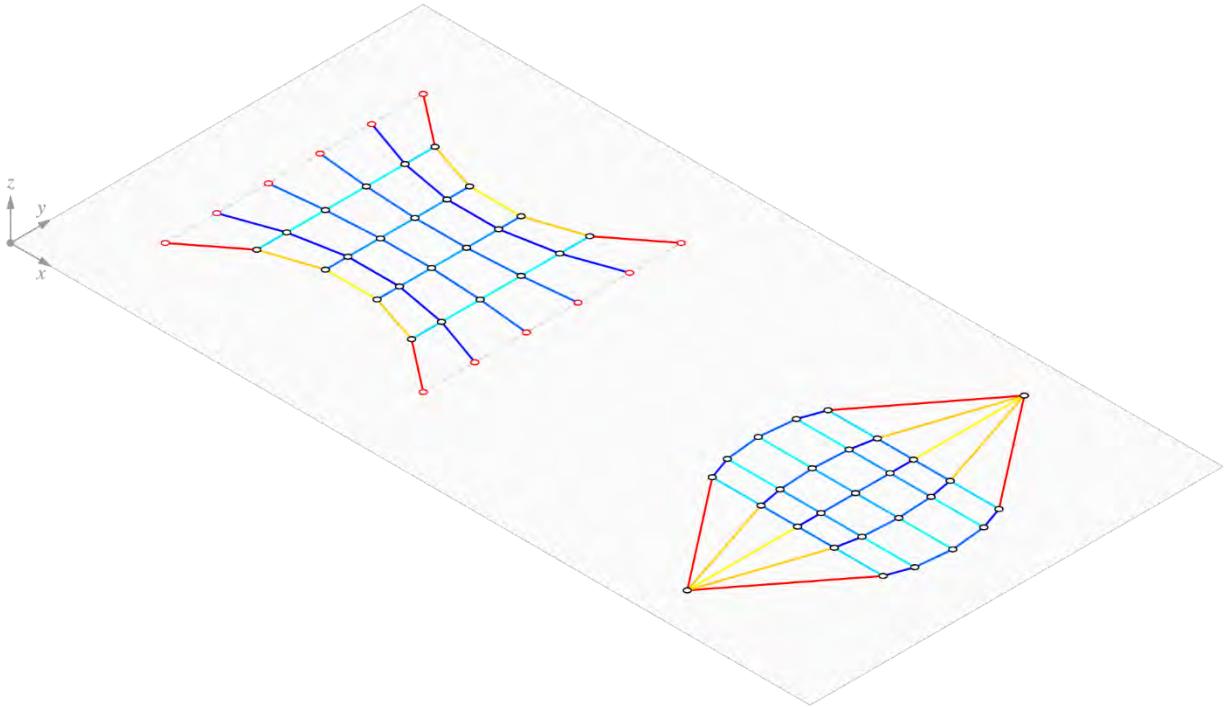
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



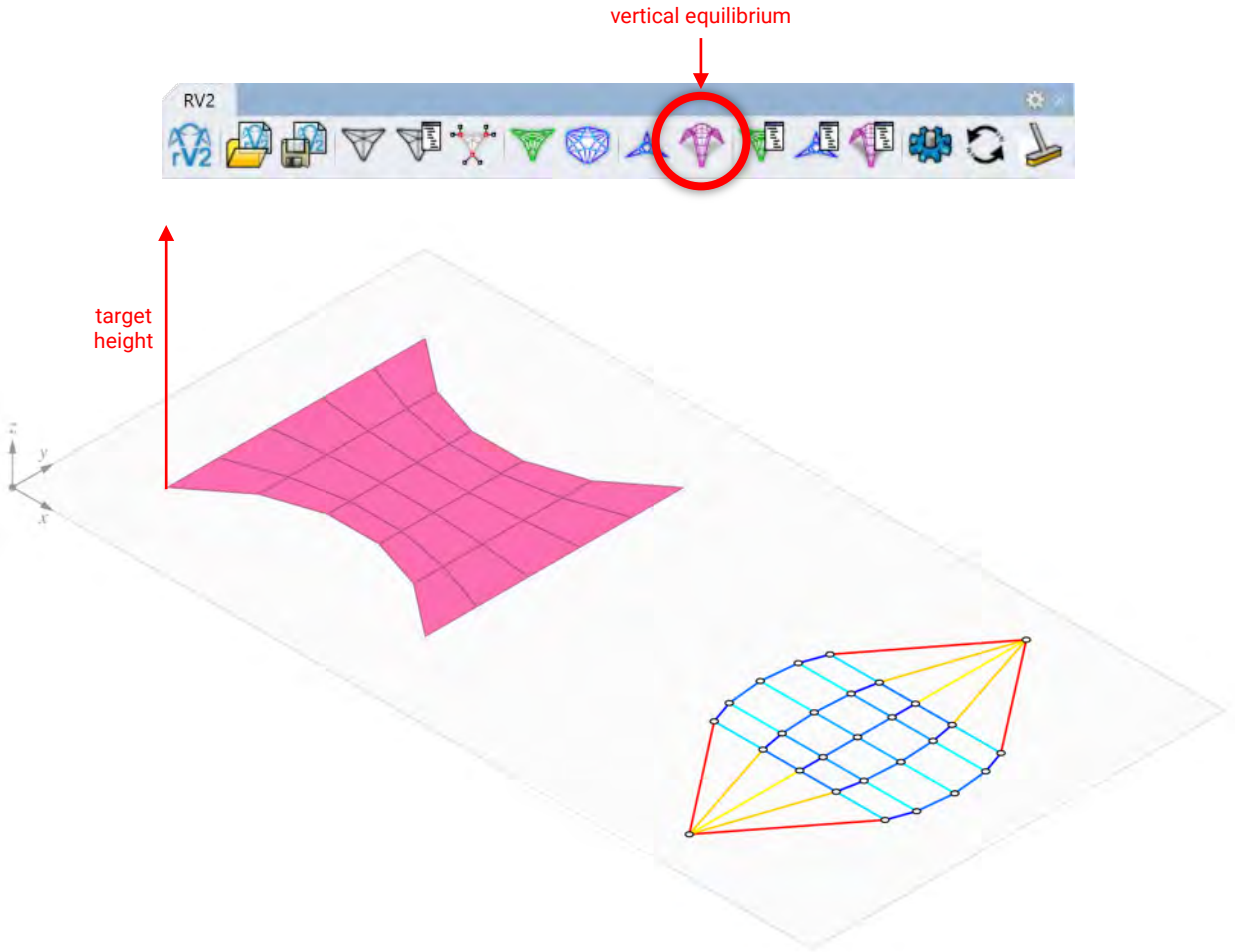
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



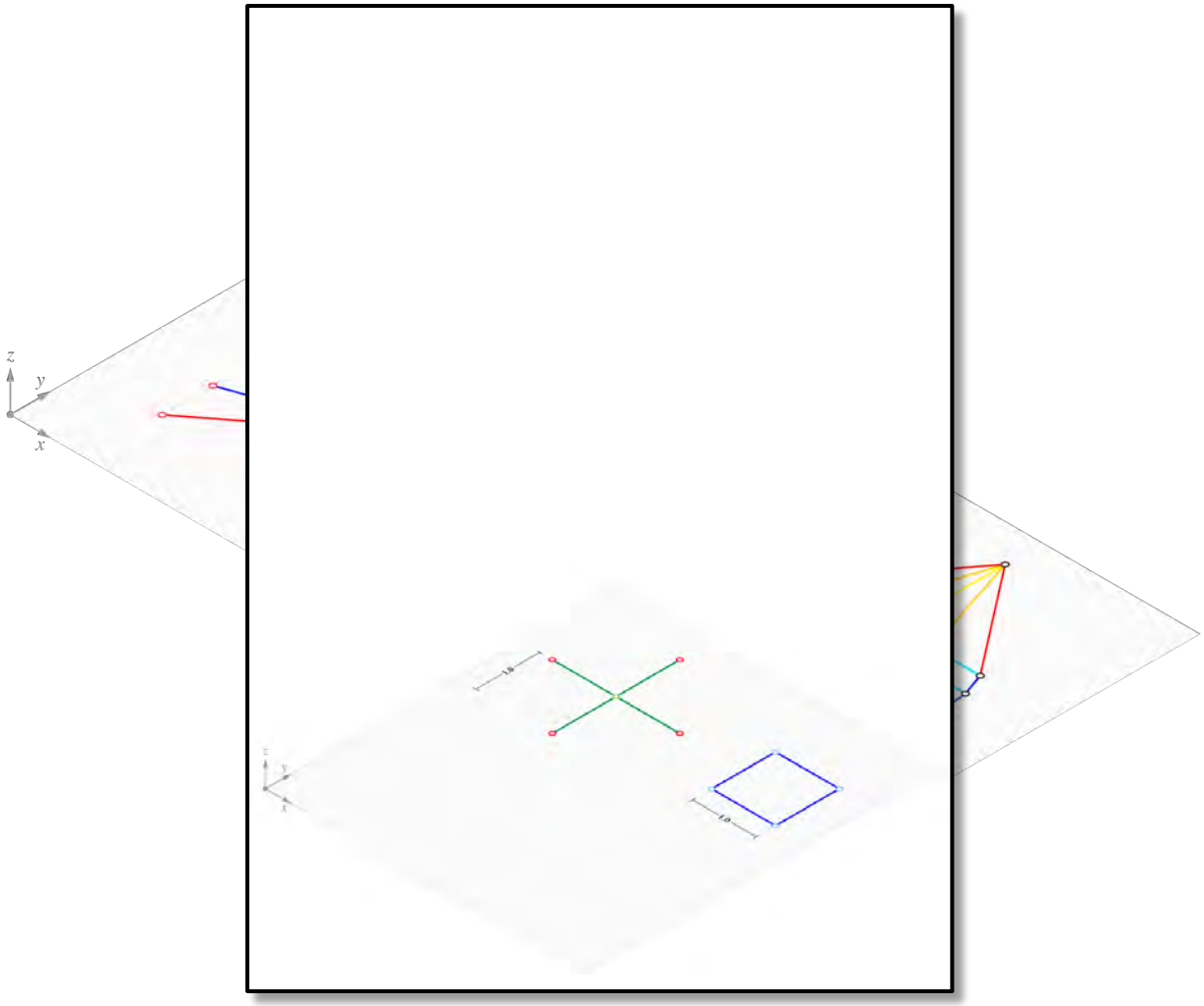
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



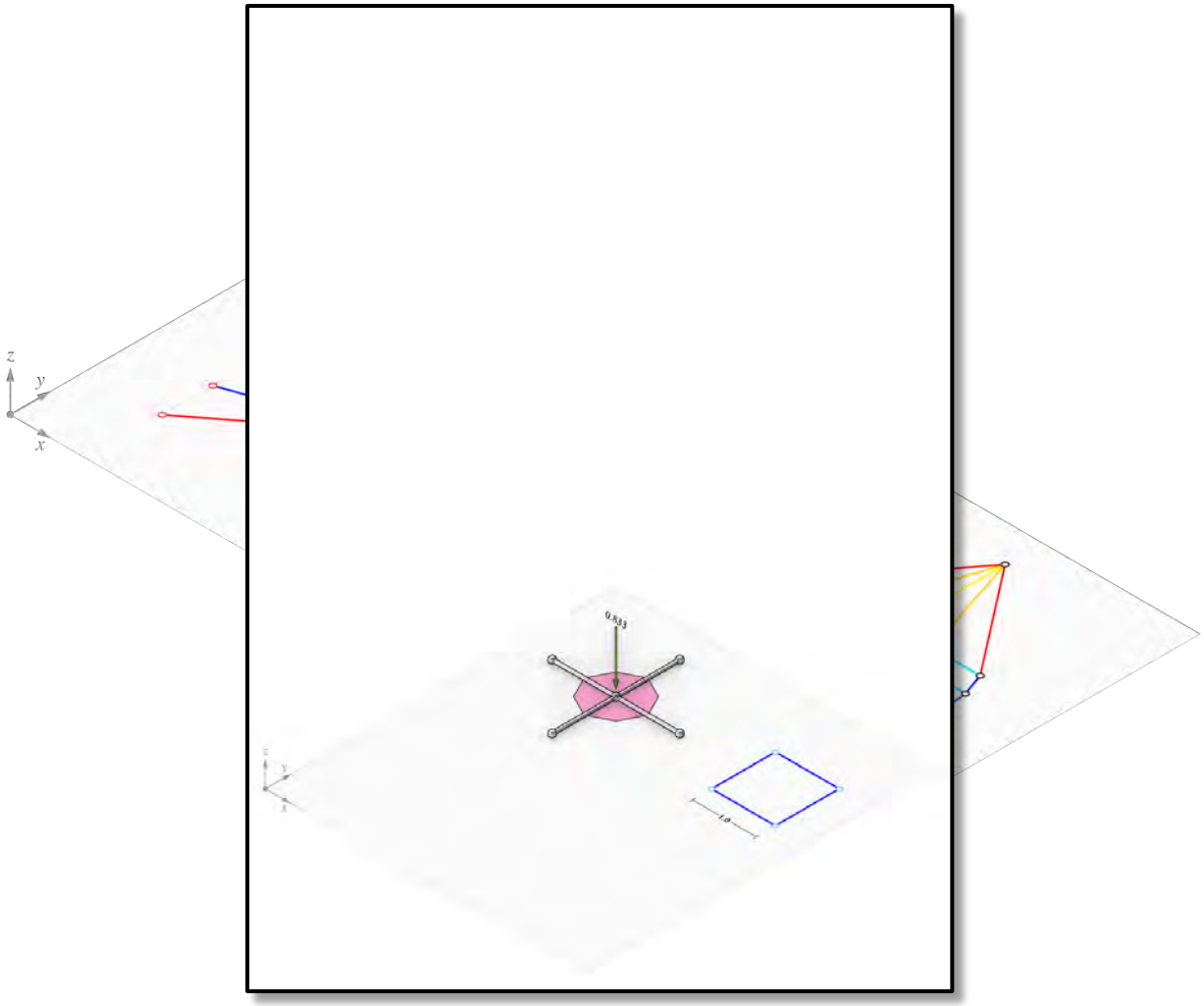
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



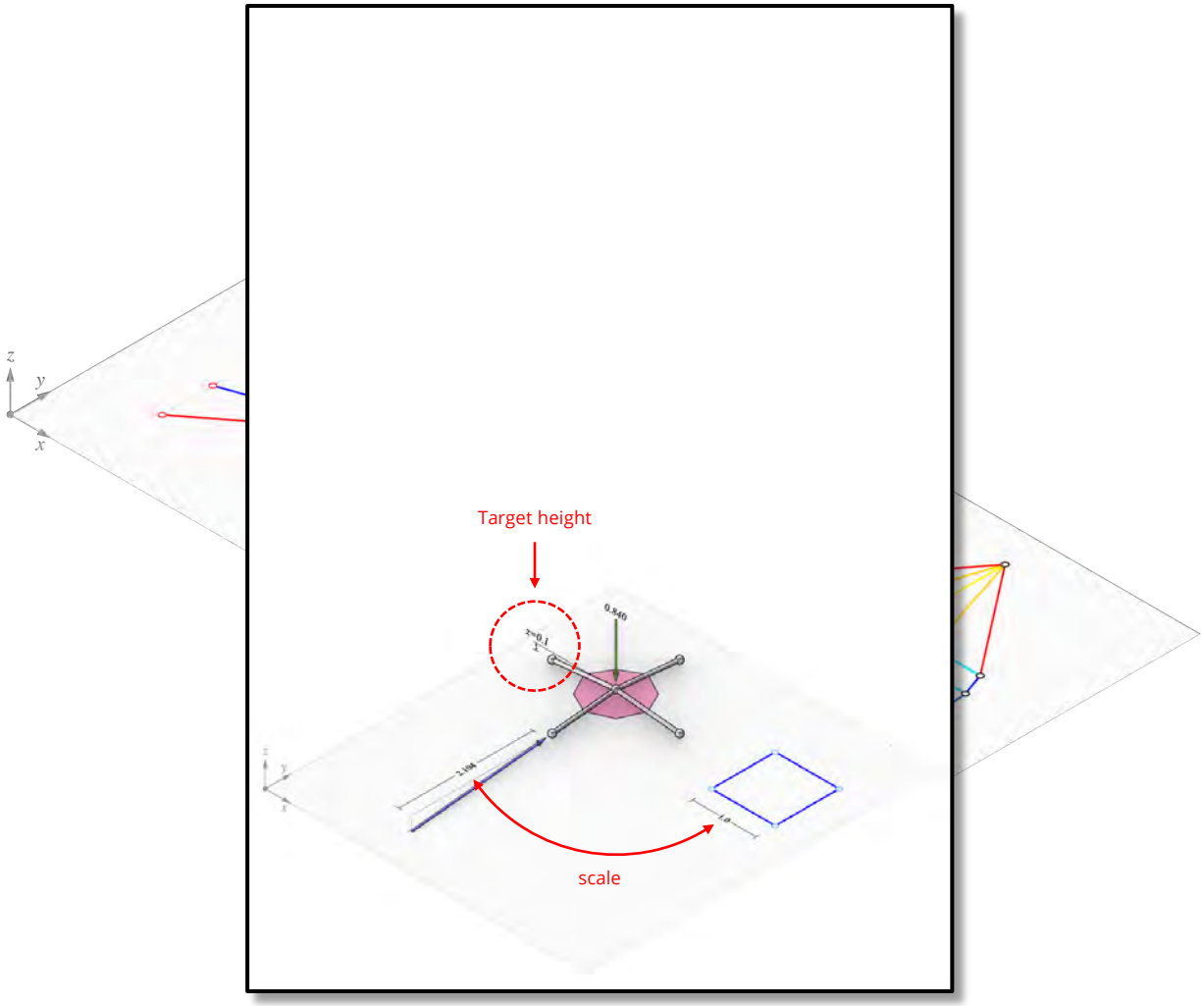
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



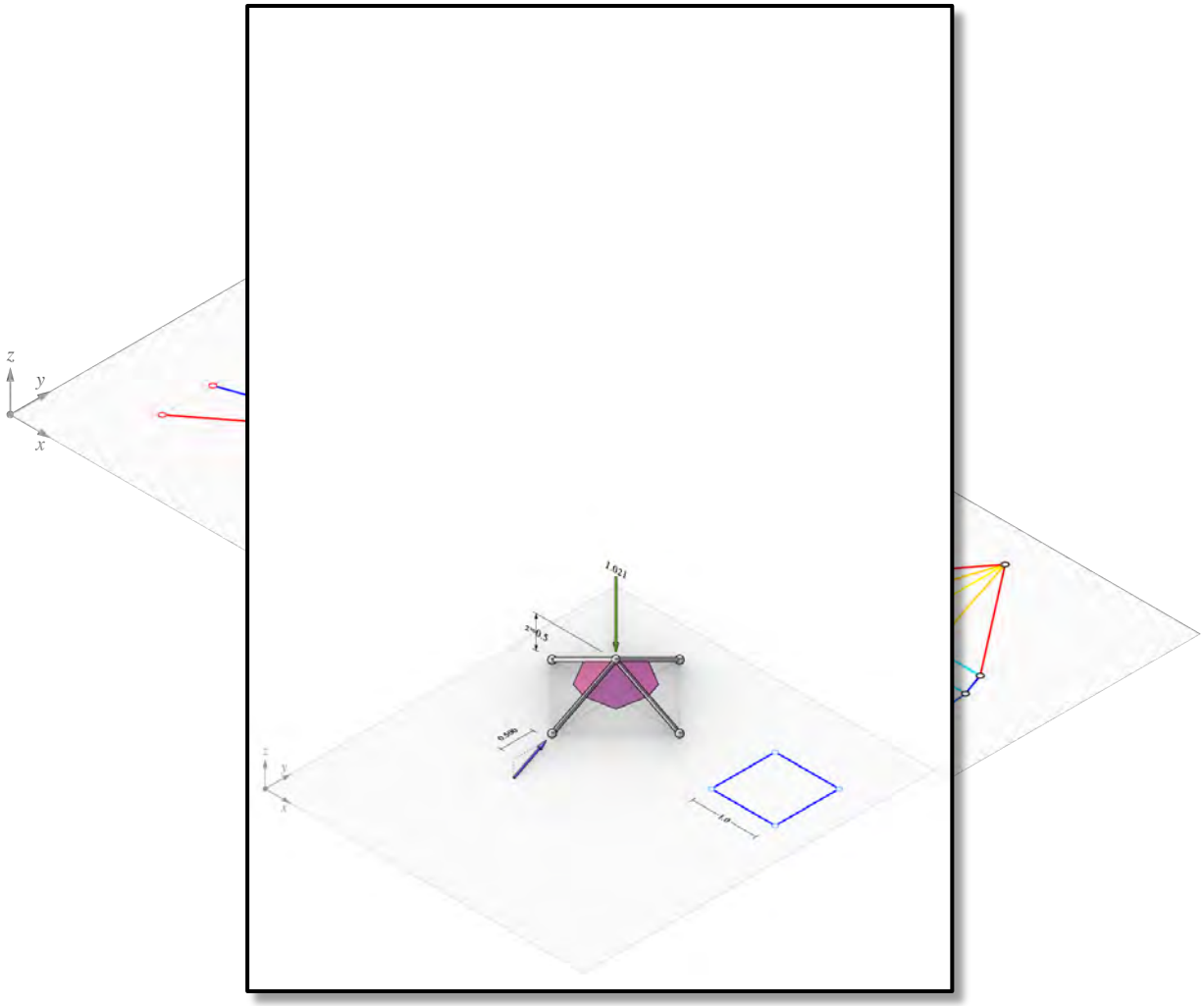
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



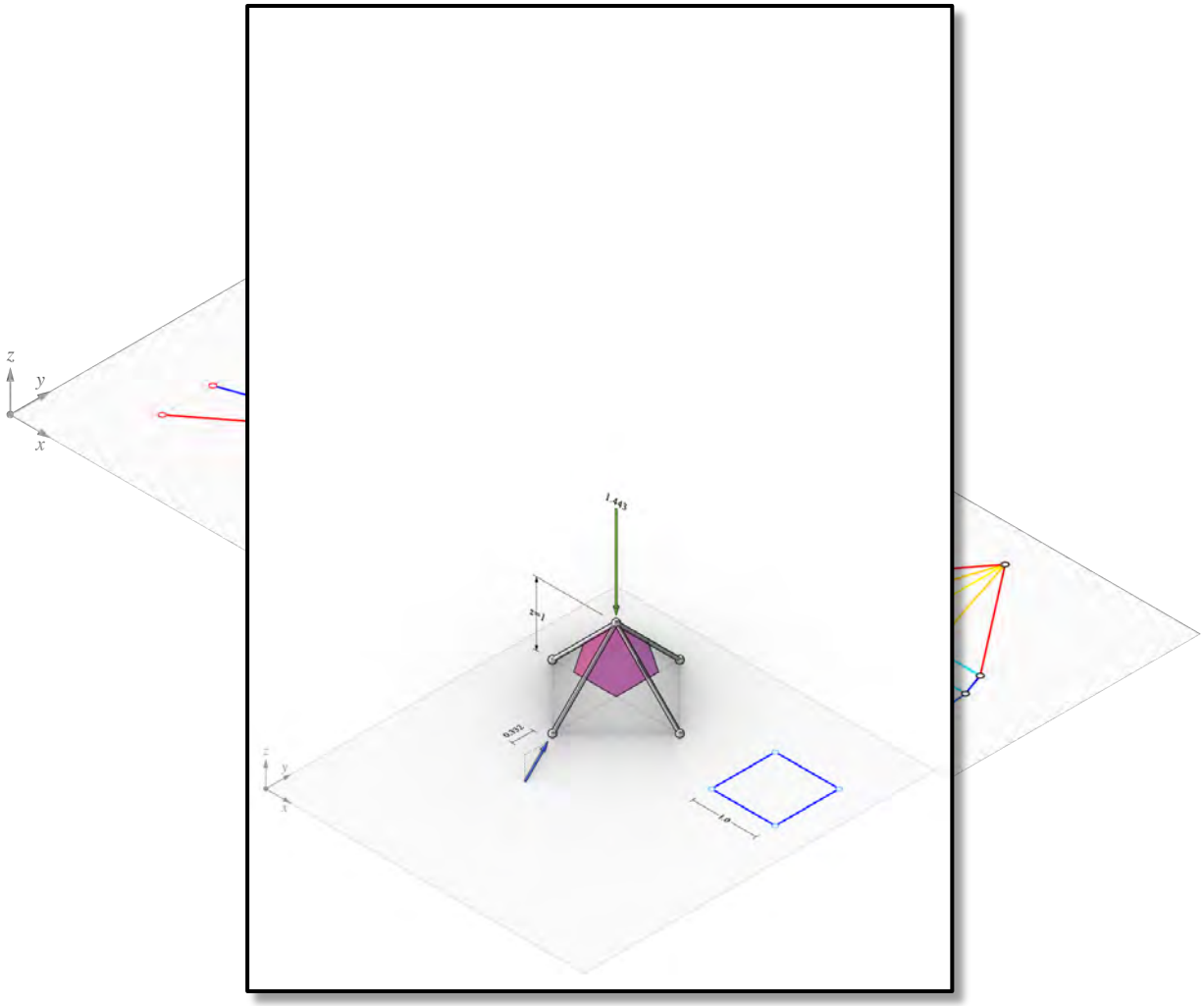
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



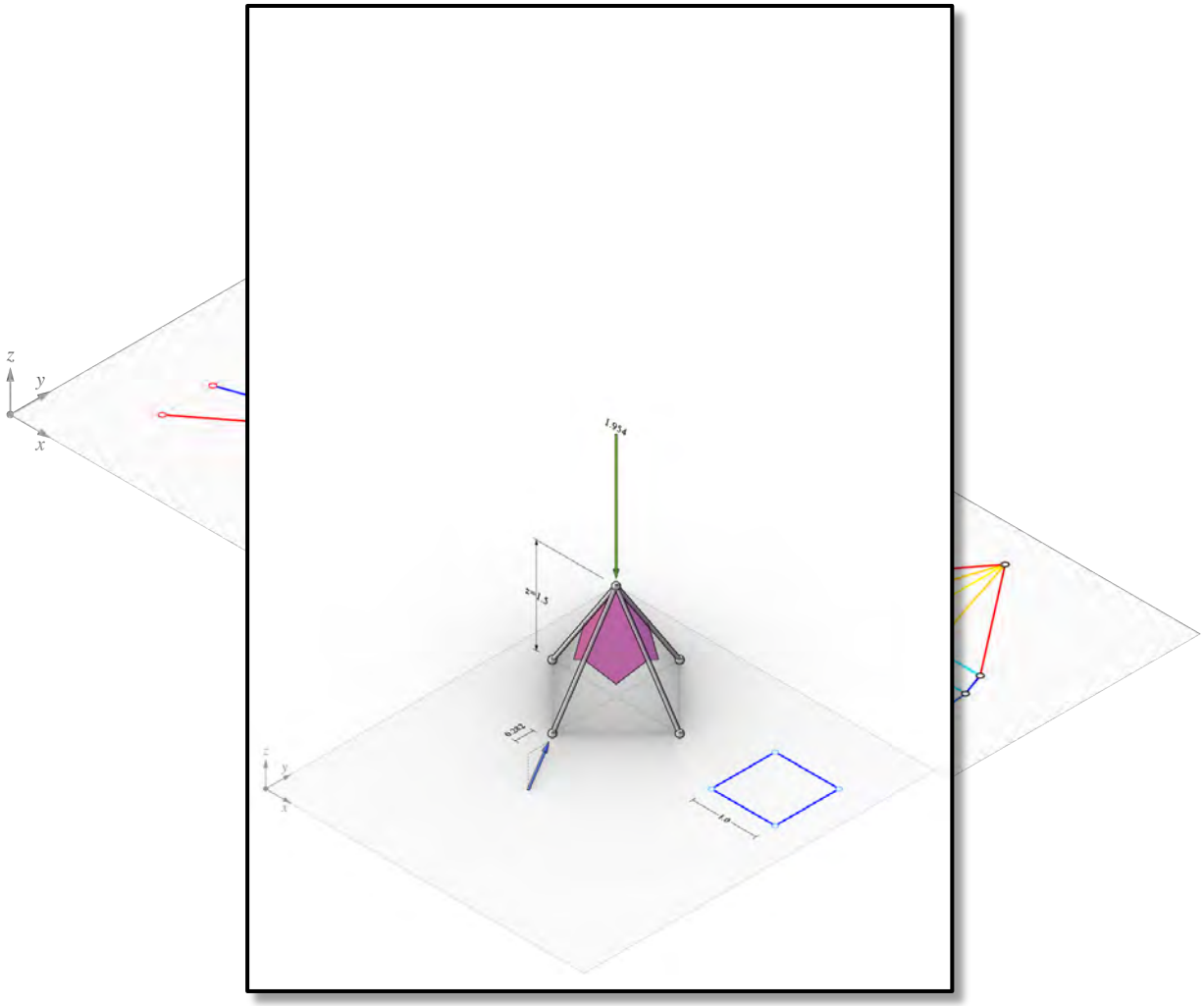
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



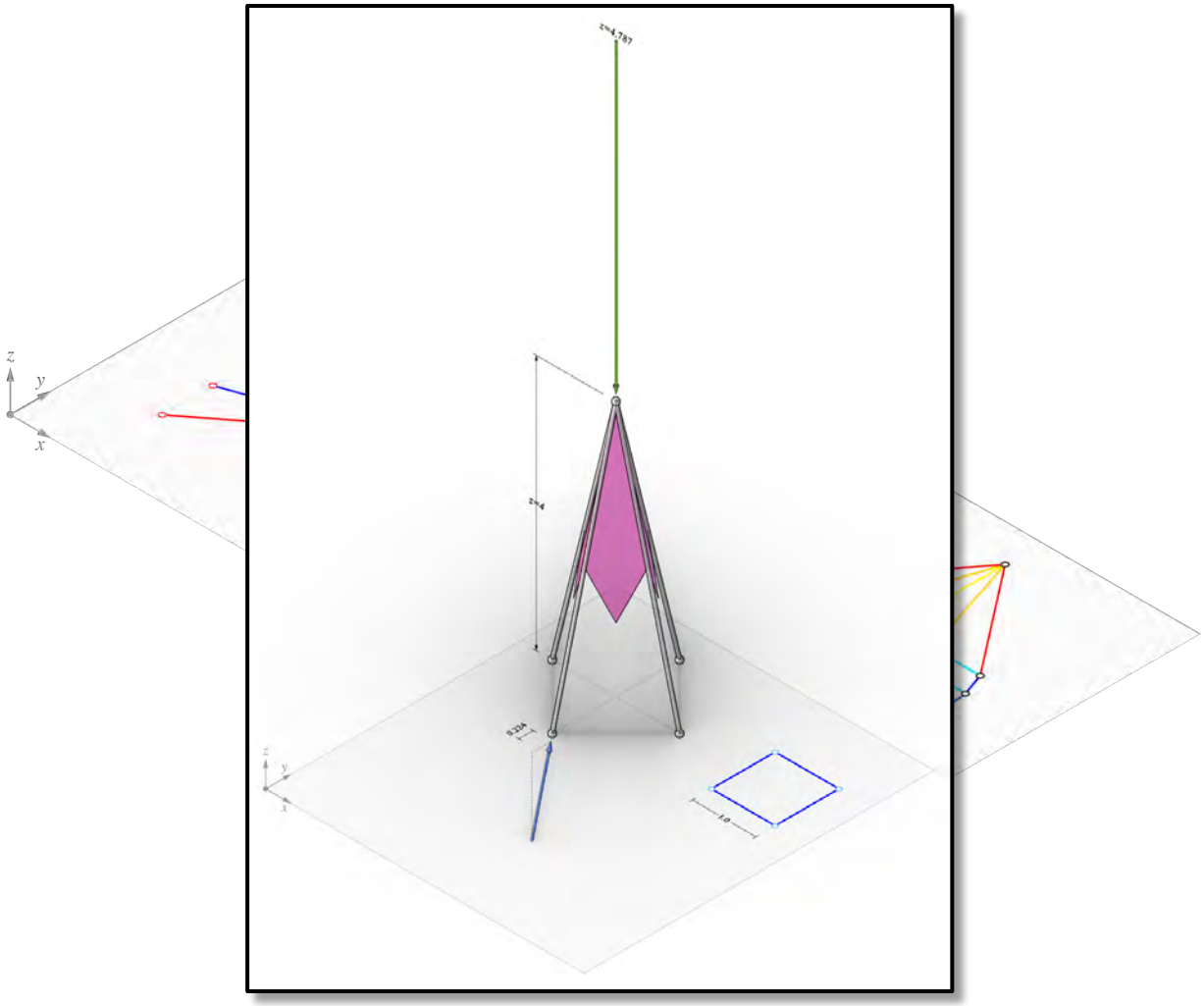
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



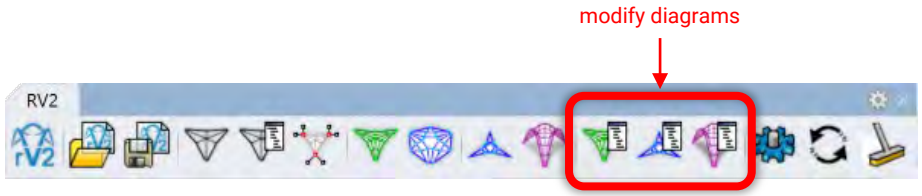
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

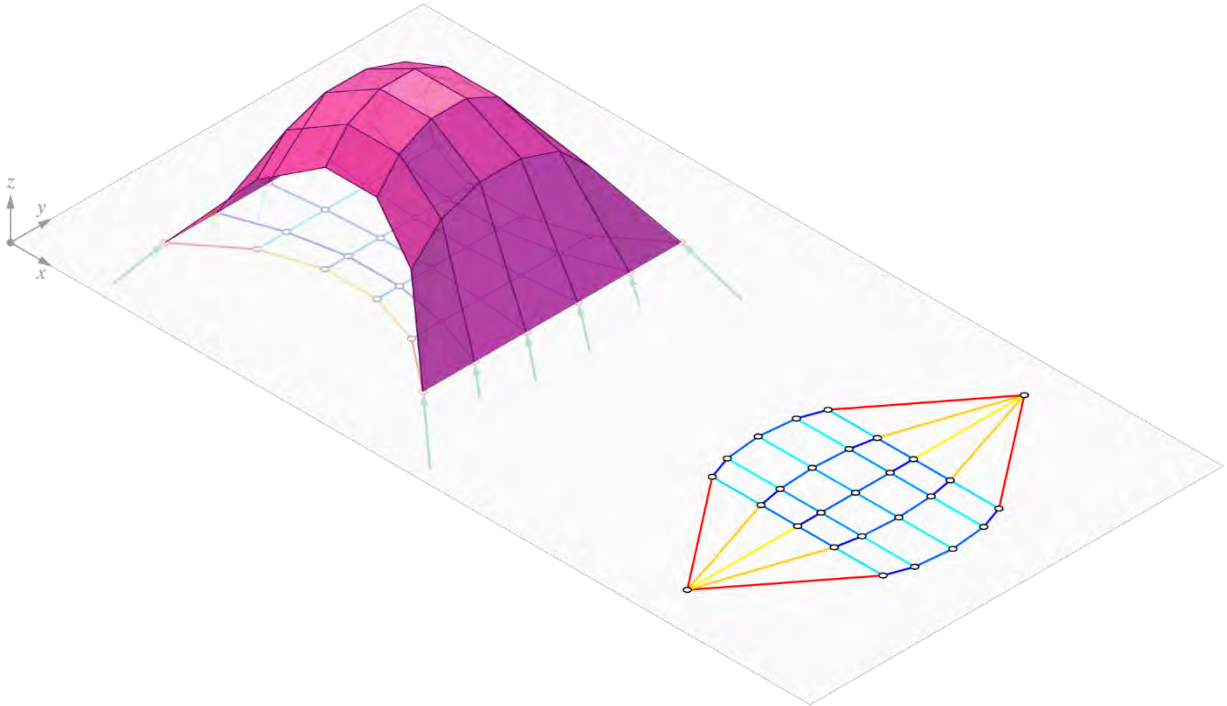


1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

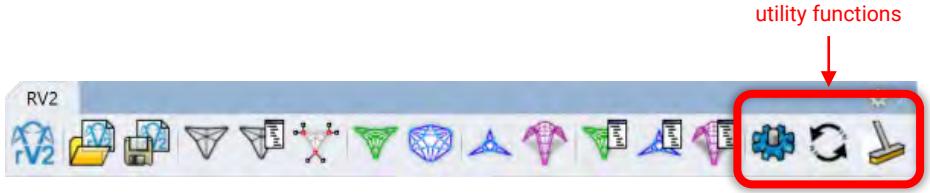


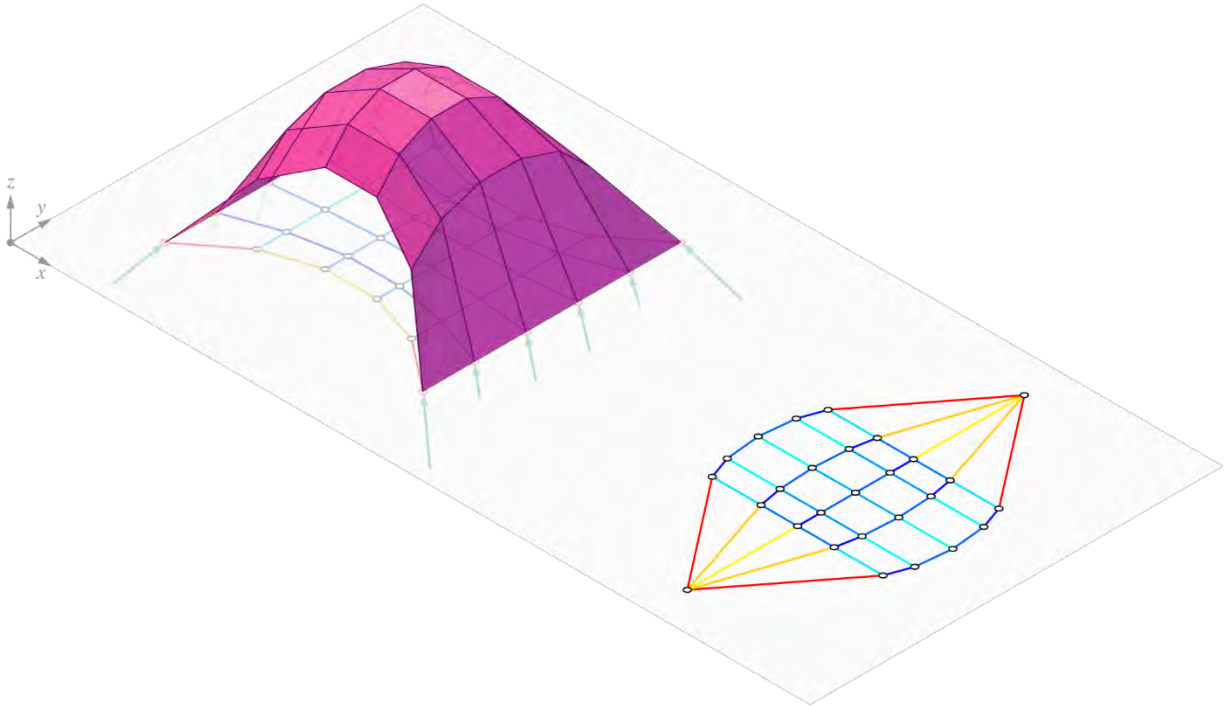
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

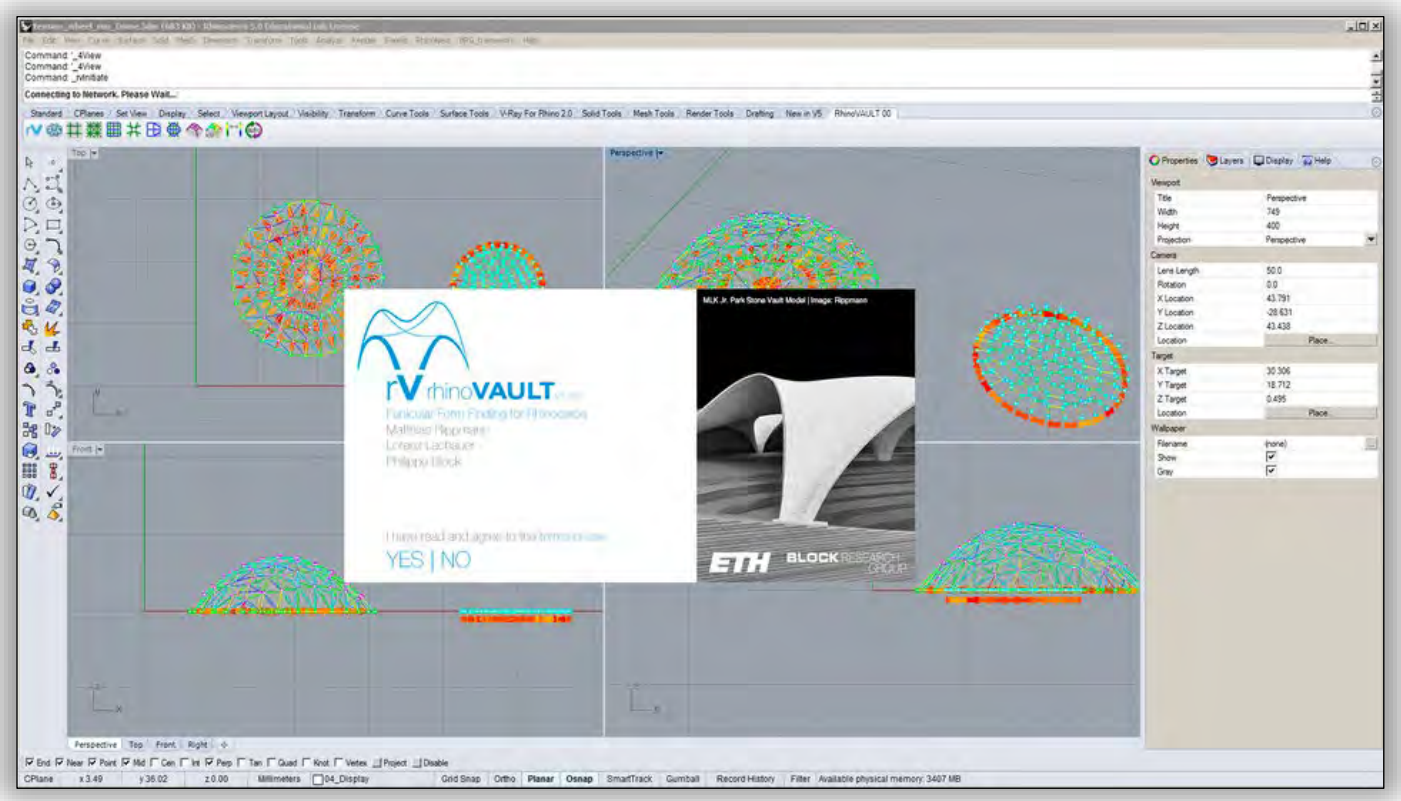




1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



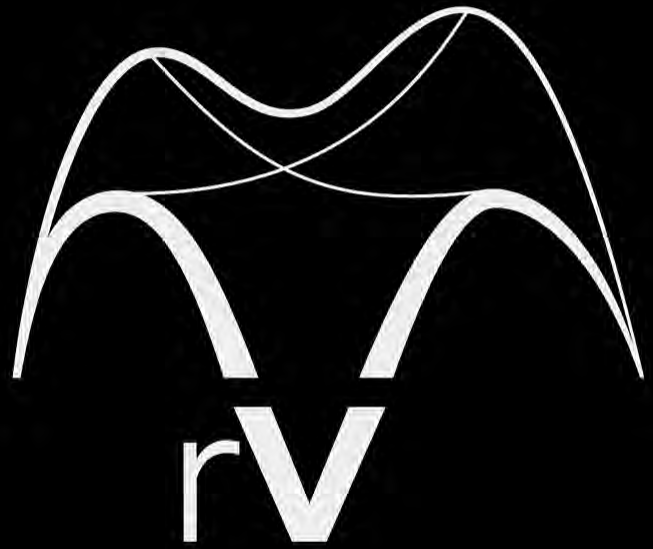




rhinoVAULT | Rippmann, Lachauer & Block (2012)

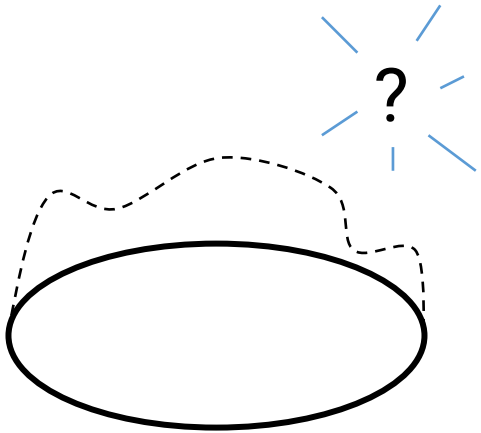
rhinoVAULT

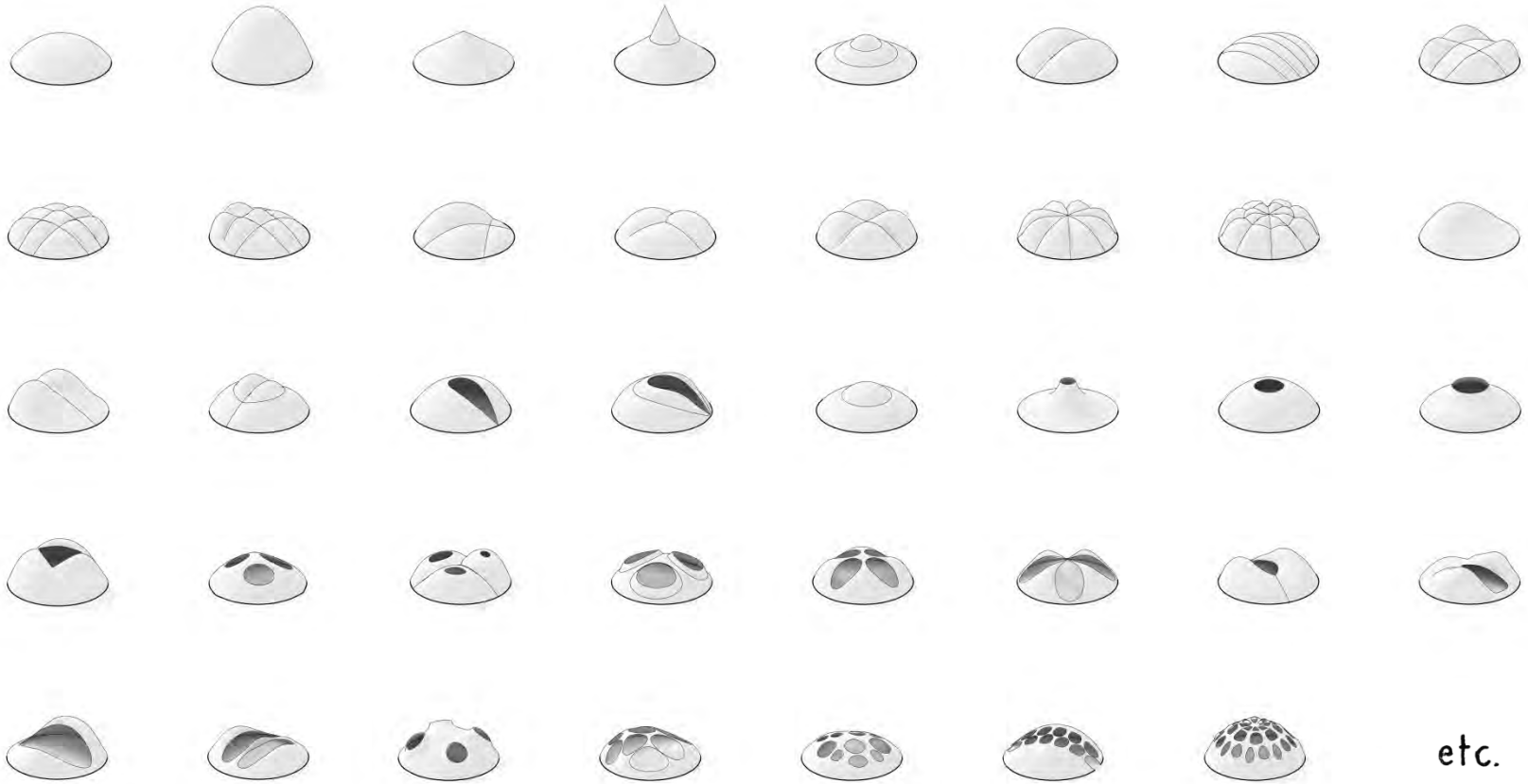
MATTHIAS RIPPMANN
LORENZ LACHAUER
PHILIPPE BLOCK

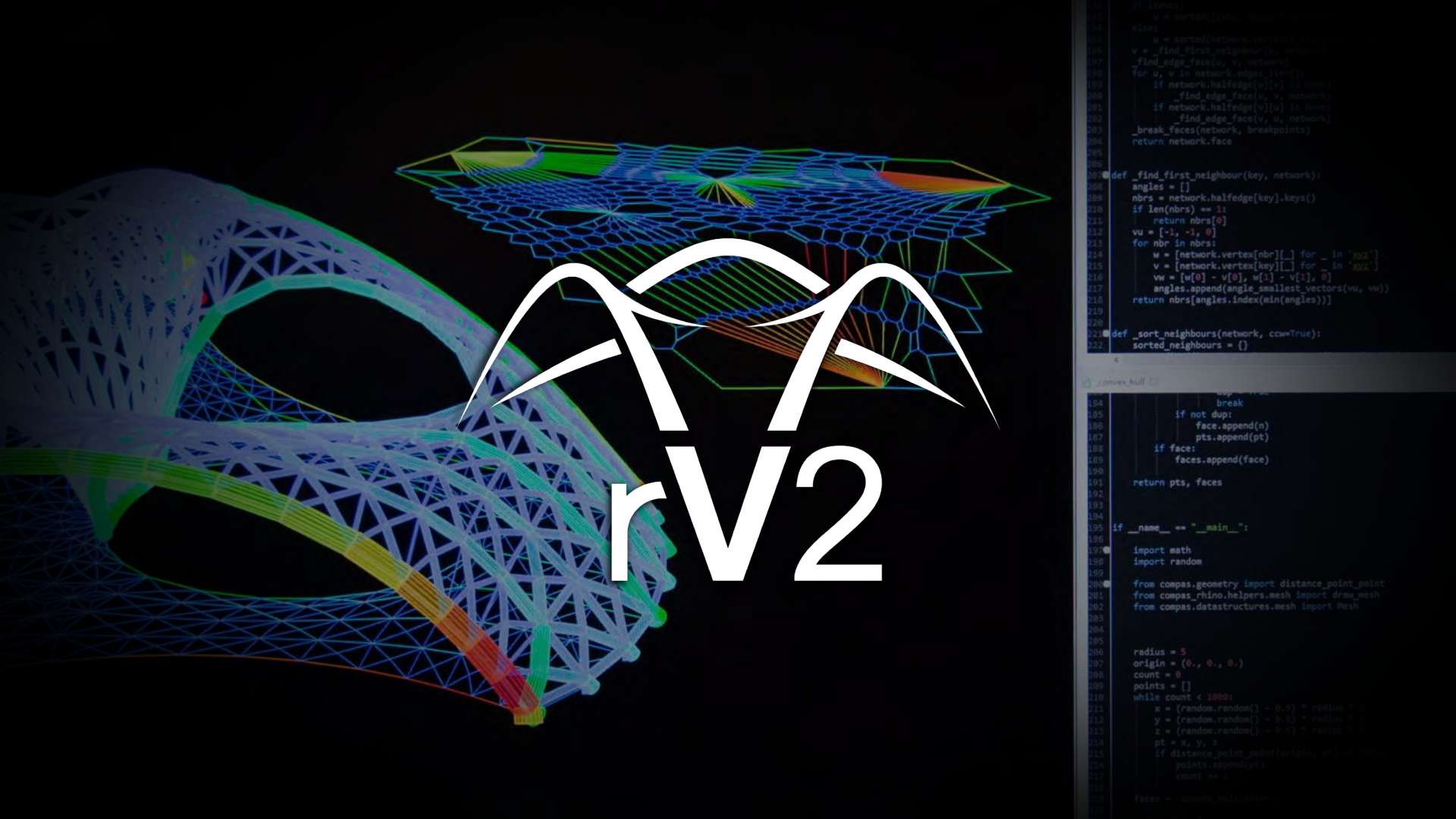


ETH Zürich

BLOCK RESEARCH GROUP



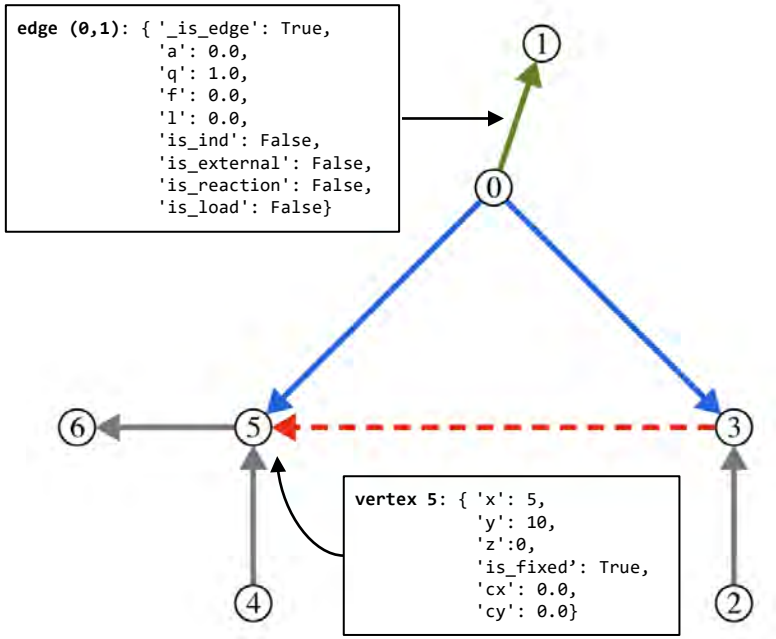

etc.


rV2

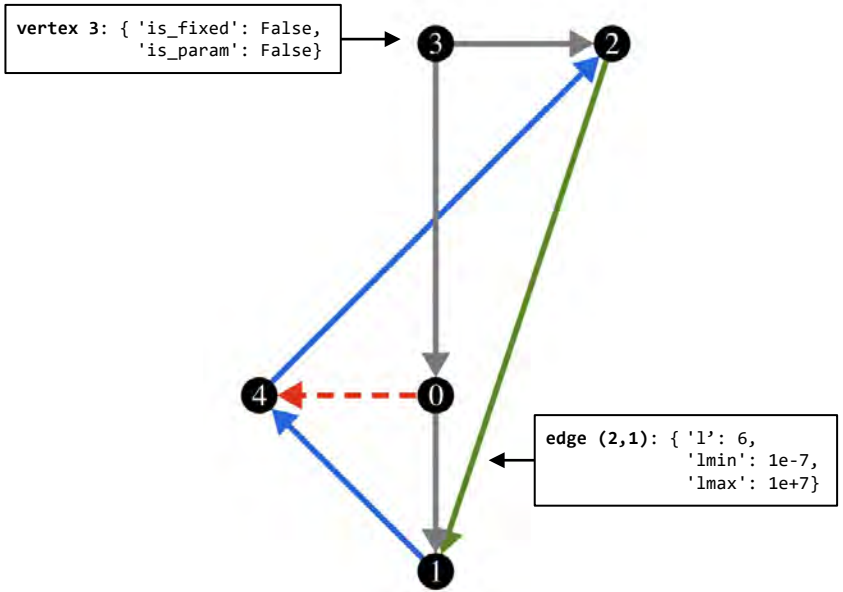
COMPAS

Open-source, Python-based framework for computational research and collaboration in architecture, engineering and digital fabrication

form diagram data



force diagram data

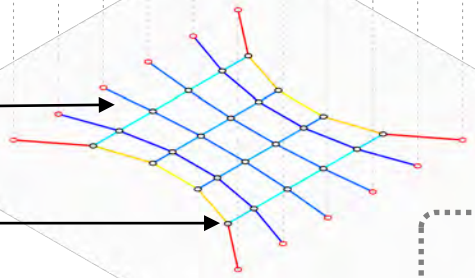


thrust diagram data

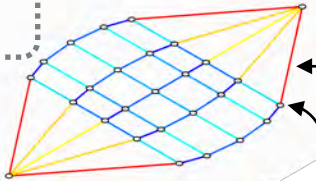
form diagram data

```
edge (u, v): {‘q’: 1.0,
              'lmin’: 0.0,
              'lmax': 1e+7,
              ...}
```

```
vertex: {‘is_fixed’: False,
         ‘is_anchor’: False,
         ‘x’: 0.0,
         ‘y’: 0.0,
         ‘z’: 0.0,
         ...}
```

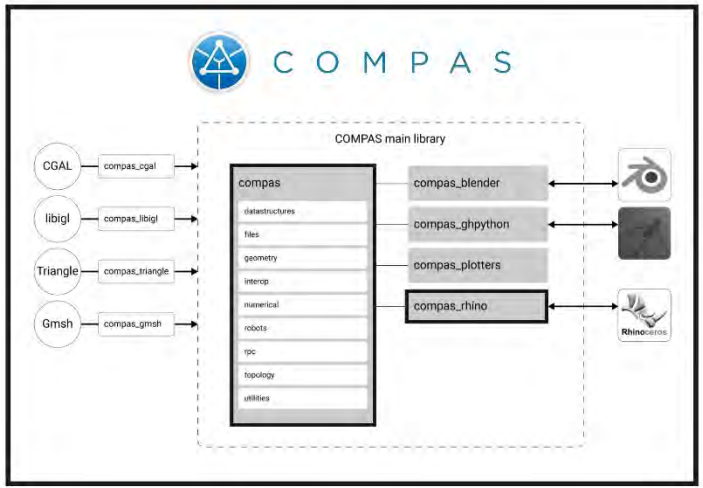


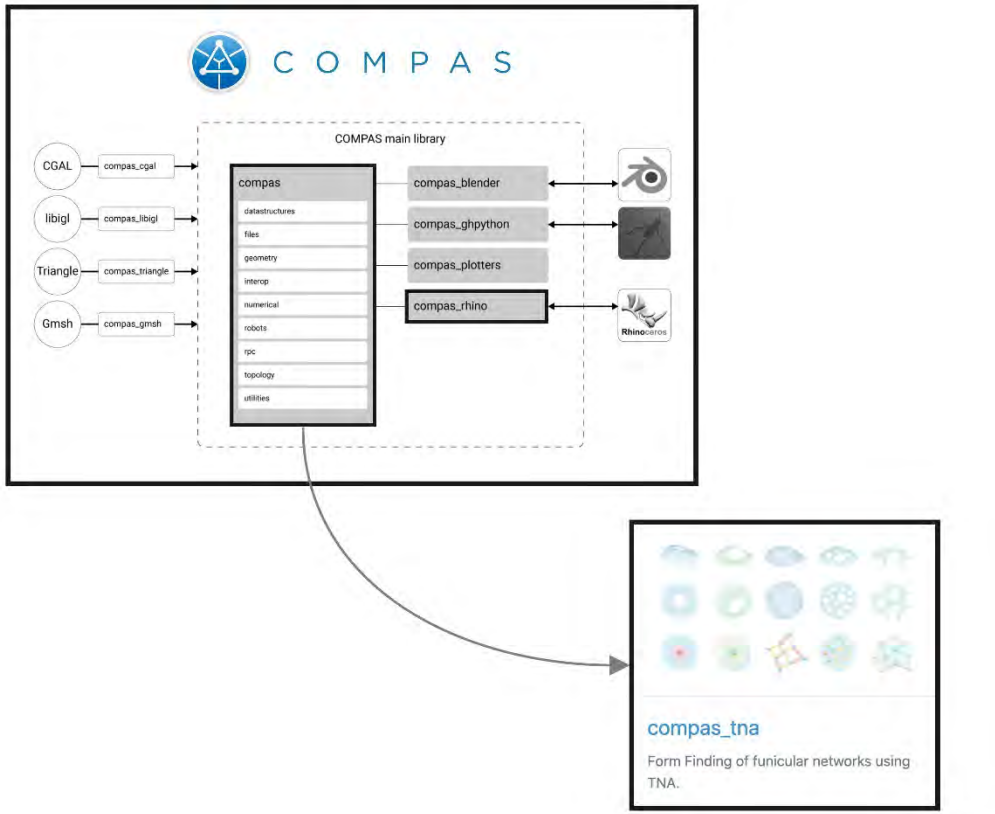
force diagram data

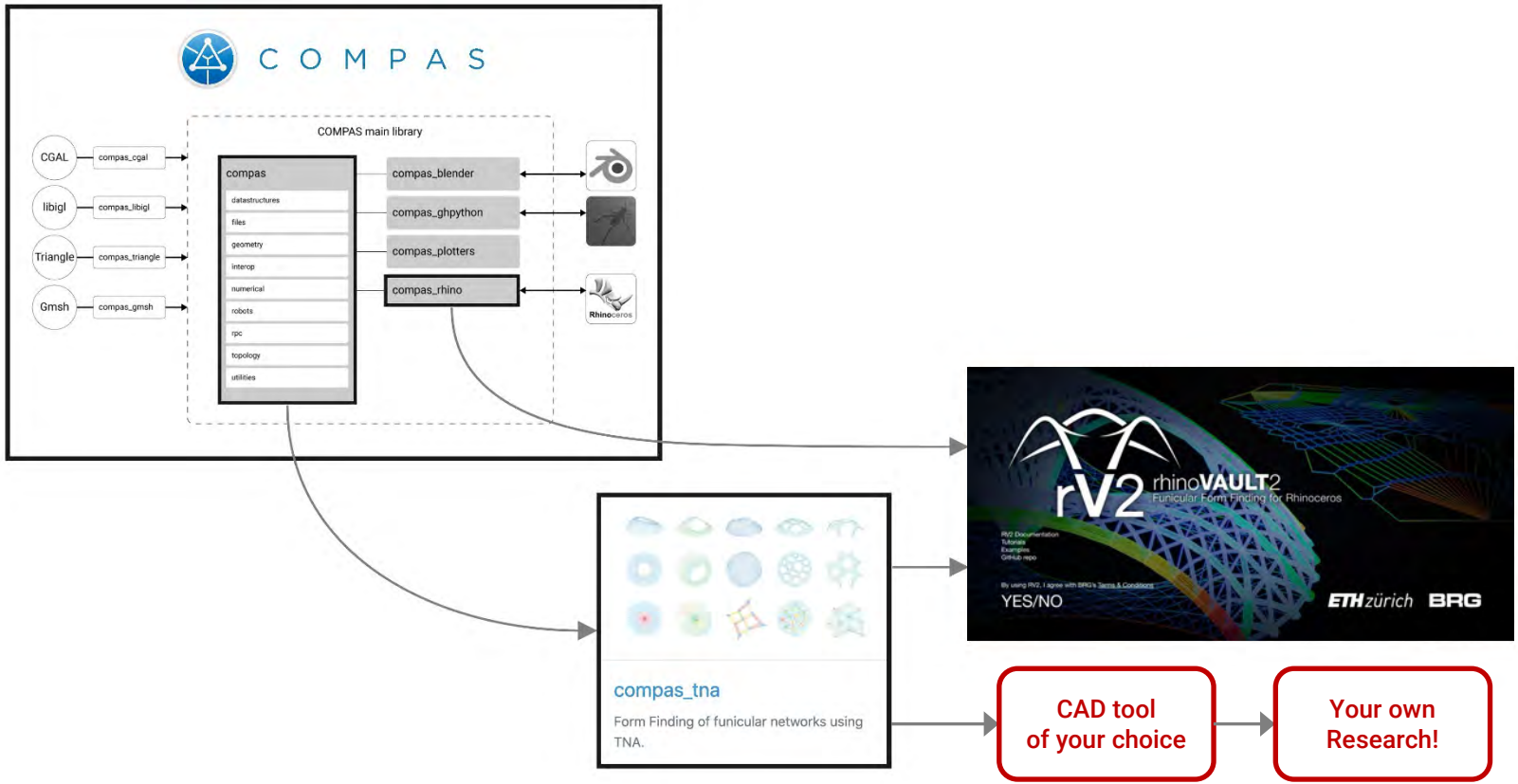


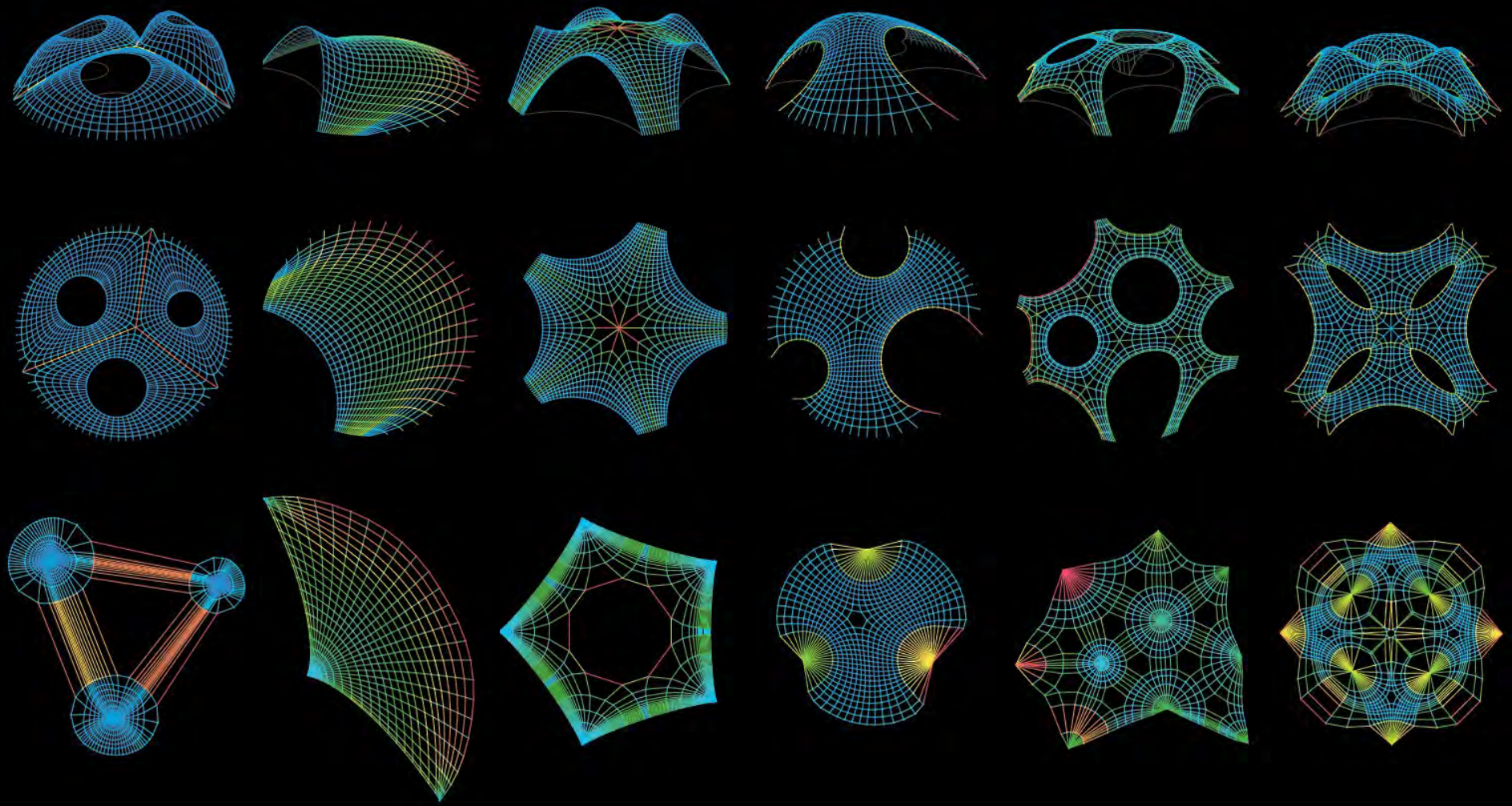
```
edge (u, v): {‘_a’: 0.0,
              'lmin’: 0.0,
              'lmax': 1e+7}
```

```
vertex: {‘is_fixed’: False,
         ‘x’: 0.0,
         ‘y’: 0.0,
         ‘z’: 0.0}
```


COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
intercop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros

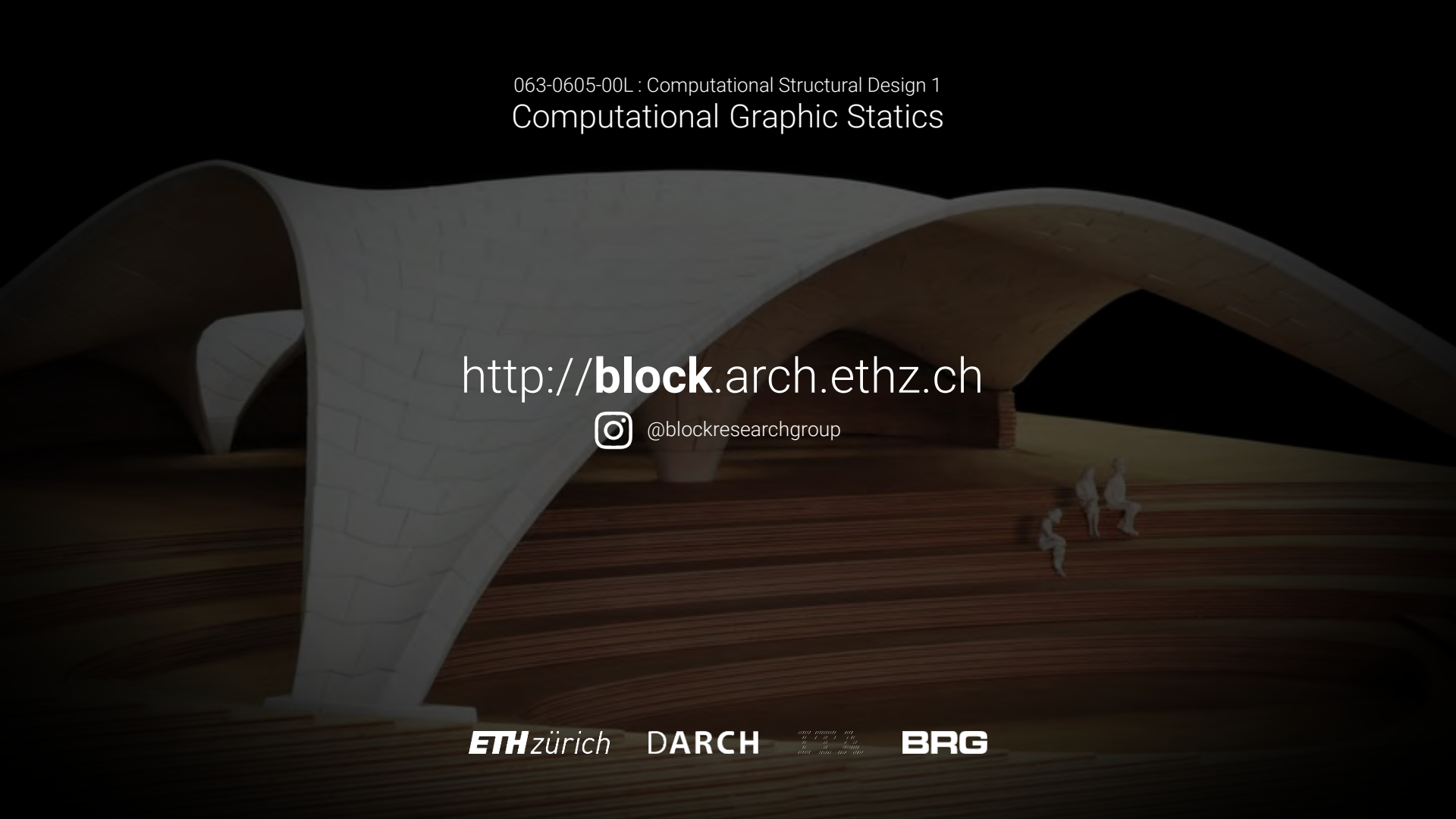

COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
intercop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros
compas_tna
Form Finding of funicular networks using TNA.


COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
interop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros
rV2
rhinoVAULT2
Funicular Form Finding for Rhinoceros
YES/NO
ETH zürich
BRG
compas_tna
Form Finding of funicular networks using TNA.
CAD tool of your choice
Your own Research!



063-0605-00L : Computational Structural Design 1

Computational Graphic Statics



http://**block**.arch.ethz.ch

@blockresearchgroup

ETH zürich DARCH BRG

compas dashboard

v0.7.1

+

RhinoVAULT 2

V1.4.3